Optimal supply & Structure detection
Complex Networks, SFI Summer School, June, 2010

Prof. Peter Dodds

Department of Mathematics & Statistics
Center for Complex Systems
Vermont Advanced Computing Center
University of Vermont

Licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Optimal supply networks

What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
Optimal supply networks

What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes

- Q: How do optimal solutions scale with system size?
Optimal supply networks

What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
Optimal supply networks

What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
Optimal supply networks

What’s the best way to distribute stuff?

▶ Stuff = medical services, energy, nutrients, people, ...

▶ Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes

▶ Q: How do optimal solutions scale with system size?
Optimal supply networks

What’s the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
  1. Distribute stuff from single source to many sinks
  2. Collect stuff coming from many sources at a single sink
  3. Distribute stuff from many sources to many sinks
  4. Redistribute stuff between many nodes
- Q: How do optimal solutions scale with system size?
Single source optimal supply

Basic Q for distribution/supply networks:

- How does flow behave given cost:
  \[ C = \sum_j I_j^\gamma Z_j \]

where
  \( I_j \) = current on link \( j \)
  \( Z_j \) = link \( j \)'s impedance?

- Example: \( \gamma = 2 \) for electrical networks.
Basic Q for distribution/supply networks:

- How does flow behave given cost:
  \[ C = \sum_j I_j^\gamma Z_j \]

  where
  \( I_j \) = current on link \( j \)
  and
  \( Z_j \) = link \( j \)'s impedance?
- Example: \( \gamma = 2 \) for electrical networks.
Single source optimal supply

(a) $\gamma > 1$: **Braided** (bulk) flow
(b) $\gamma < 1$: Local minimum: **Branching** flow
(c) $\gamma < 1$: Global minimum: **Branching** flow

From Bohn and Magnasco $^[3]$  
See also Banavar et al. $^[1]$
Single source optimal supply

Optimal paths related to transport (Monge) problems:

Xia (2003) [24]
Growing networks:

**Figure 1.** $\alpha = 0.6, \beta = 0.5$

- $\alpha = 0.6, \beta = 0.5, \varepsilon = 2$
- $\alpha = 0.6, \beta = 0.5, \varepsilon = 3$
- $\alpha = 0.6, \beta = 0.5, \varepsilon = 4$
- $\alpha = 0.6, \beta = 0.5, \varepsilon = 5$

Xia (2007) [23]
Growing networks:

**Figure 3. A maple leaf**

Xia (2007) [23]
Single source optimal supply

An immensely controversial issue...

- The form of river networks and blood networks: optimal or not? [22, 2, 7]

Two observations:

- Self-similar networks appear everywhere in nature for single source supply/single sink collection.
- Real networks differ in details of scaling but reasonably agree in scaling relations.
Single source optimal supply

An immensely controversial issue...

▶ The form of river networks and blood networks: optimal or not? [22, 2, 7]

Two observations:

▶ Self-similar networks appear everywhere in nature for single source supply/single sink collection.
▶ Real networks differ in details of scaling but reasonably agree in scaling relations.
Single source optimal supply

An immensely controversial issue...

► The form of river networks and blood networks: optimal or not? [22, 2, 7]

Two observations:

► Self-similar networks appear everywhere in nature for single source supply/single sink collection.
► Real networks differ in details of scaling but reasonably agree in scaling relations.
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:

  $\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$

  where $\delta$ is the Kronecker delta.
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:
  \[ \omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2} \]
  where $\delta$ is the Kronecker delta.
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
  - Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
  - If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
  - Simple rule:
    $$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$
    where $\delta$ is the Kronecker delta.
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:
  $$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$
  where $\delta$ is the Kronecker delta.

![Mississippi River with order labels](source=/data6/dodds/work/rivers/dems/mississippi/figures/figorder_paths_mispi10.ps)
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:
  $$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1,\omega_2}$$
  where $\delta$ is the Kronecker delta.
Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:
  \[ \omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2} \]
  where $\delta$ is the Kronecker delta.
Horton’s laws in the real world:

**The Mississippi**

- $n_\omega$ (number of streams)
- $a_\omega$ (area, sq km)
- $l_\omega$ (length, km)

**The Nile**

- $n_\omega$
- $a_\omega$ (sq km)
- $l_\omega$ (km)

**The Amazon**

- $n_\omega$
- $a_\omega$ (sq km)
- $l_\omega$ (km)
Many scaling laws, many connections

<table>
<thead>
<tr>
<th>relation:</th>
<th>scaling relation/parameter: $[6]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell \sim L^d$</td>
<td>$d$</td>
</tr>
<tr>
<td>$T_k = T_1 (R_T)^{k^{-1}}$</td>
<td>$T_1 = R_n - R_s - 2 + 2R_s/R_n$</td>
</tr>
<tr>
<td>$n_ω/n_{ω+1} = R_n$</td>
<td>$R_n$</td>
</tr>
<tr>
<td>$\tilde{a}_{ω+1}/\tilde{a}_ω = R_a$</td>
<td>$R_a = R_n$</td>
</tr>
<tr>
<td>$\tilde{ℓ}_{ω+1}/\tilde{ℓ}_ω = R_ℓ$</td>
<td>$R_ℓ = R_s$</td>
</tr>
<tr>
<td>$\ell \sim a^h$</td>
<td>$h = \log R_s/\log R_n$</td>
</tr>
<tr>
<td>$a \sim L^D$</td>
<td>$D = d/h$</td>
</tr>
<tr>
<td>$L_⊥ \sim L^H$</td>
<td>$H = d/h - 1$</td>
</tr>
<tr>
<td>$P(a) \sim a^{-\tau}$</td>
<td>$\tau = 2 - h$</td>
</tr>
<tr>
<td>$P(ℓ) \sim ℓ^{-γ}$</td>
<td>$γ = 1/h$</td>
</tr>
<tr>
<td>$Λ \sim a^β$</td>
<td>$β = 1 + h$</td>
</tr>
<tr>
<td>$λ \sim L^φ$</td>
<td>$φ = d$</td>
</tr>
</tbody>
</table>

Only 3 parameters are independent... $[6]$
Reported parameter values: [6]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Real networks:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_n$</td>
<td>3.0–5.0</td>
</tr>
<tr>
<td>$R_a$</td>
<td>3.0–6.0</td>
</tr>
<tr>
<td>$R_l = R_T$</td>
<td>1.5–3.0</td>
</tr>
<tr>
<td>$T_1$</td>
<td>1.0–1.5</td>
</tr>
<tr>
<td>$d$</td>
<td>1.1 ± 0.01</td>
</tr>
<tr>
<td>$D$</td>
<td>1.8 ± 0.1</td>
</tr>
<tr>
<td>$h$</td>
<td>0.50–0.70</td>
</tr>
<tr>
<td>$\tau$</td>
<td>1.43 ± 0.05</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>1.8 ± 0.1</td>
</tr>
<tr>
<td>$H$</td>
<td>0.75–0.80</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.50–0.70</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>1.05 ± 0.05</td>
</tr>
</tbody>
</table>
## Data from real blood networks

<table>
<thead>
<tr>
<th>Network</th>
<th>$R_n$</th>
<th>$R_r^{-1}$</th>
<th>$R_\ell^{-1}$</th>
<th>$-\frac{\ln R_r}{\ln R_n}$</th>
<th>$-\frac{\ln R_\ell}{\ln R_n}$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>West <em>et al.</em></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.5</td>
<td>0.33</td>
<td>0.75</td>
</tr>
<tr>
<td>rat (PAT)</td>
<td>2.76</td>
<td>1.58</td>
<td>1.60</td>
<td>0.45</td>
<td>0.46</td>
<td>0.73</td>
</tr>
<tr>
<td>cat (PAT)</td>
<td>3.67</td>
<td>1.71</td>
<td>1.78</td>
<td>0.41</td>
<td>0.44</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Turcotte <em>et al.</em> [21])</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dog (PAT)</td>
<td>3.69</td>
<td>1.67</td>
<td>1.52</td>
<td>0.39</td>
<td>0.32</td>
<td>0.90</td>
</tr>
<tr>
<td>pig (LCX)</td>
<td>3.57</td>
<td>1.89</td>
<td>2.20</td>
<td>0.50</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>pig (RCA)</td>
<td>3.50</td>
<td>1.81</td>
<td>2.12</td>
<td>0.47</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>pig (LAD)</td>
<td>3.51</td>
<td>1.84</td>
<td>2.02</td>
<td>0.49</td>
<td>0.56</td>
<td>0.65</td>
</tr>
<tr>
<td>human (PAT)</td>
<td>3.03</td>
<td>1.60</td>
<td>1.49</td>
<td>0.42</td>
<td>0.36</td>
<td>0.83</td>
</tr>
<tr>
<td>human (PAT)</td>
<td>3.36</td>
<td>1.56</td>
<td>1.49</td>
<td>0.37</td>
<td>0.33</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Animal power

Fundamental biological and ecological constraint:

\[ P = c M^\alpha \]

- \( P \) = basal metabolic rate
- \( M \) = organismal body mass
Animal power

Fundamental biological and ecological constraint:

\[ P = c M^\alpha \]

\( P = \) basal metabolic rate
\( M = \) organismal body mass
History

1964: Troon, Scotland: 3rd symposium on energy metabolism. $\alpha = 3/4$ made official ...
1964: Troon, Scotland: 3rd symposium on energy metabolism. $\alpha = 3/4$ made official ... ... 29 to zip.
Some data on metabolic rates

$B = 0.026\, M^{0.668}$

- Heusner’s data (1991)\(^{[11]}\)
- 391 Mammals
- blue line: 2/3
- red line: 3/4.
- $(B = P)$
Some regressions from the ground up...

<table>
<thead>
<tr>
<th>range of $M$</th>
<th>$N$</th>
<th>$\hat{\alpha}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\leq 0.1$ kg</td>
<td>167</td>
<td>$0.678 \pm 0.038$</td>
</tr>
<tr>
<td>$\leq 1$ kg</td>
<td>276</td>
<td>$0.662 \pm 0.032$</td>
</tr>
<tr>
<td>$\leq 10$ kg</td>
<td>357</td>
<td>$0.668 \pm 0.019$</td>
</tr>
<tr>
<td>$\leq 25$ kg</td>
<td>366</td>
<td>$0.669 \pm 0.018$</td>
</tr>
<tr>
<td>$\leq 35$ kg</td>
<td>371</td>
<td>$0.675 \pm 0.018$</td>
</tr>
<tr>
<td>$\leq 350$ kg</td>
<td>389</td>
<td>$0.706 \pm 0.016$</td>
</tr>
<tr>
<td>$\leq 3670$ kg</td>
<td>391</td>
<td>$0.710 \pm 0.021$</td>
</tr>
</tbody>
</table>
Analysis of residuals—p-values—mammals:

- (a) $M < 3.2$ kg
- (b) $M < 10$ kg
- (c) $M < 32$ kg
- (d) all mammals.

For a-d, $p_{2/3} > 0.05$ and $p_{3/4} \ll 10^{-4}$. 
Analysis of residuals—p-values—birds:

- (a) $M < 0.1$ kg
- (b) $M < 1$ kg
- (c) $M < 10$ kg
- (d) all birds.

For a-d, $p_{2/3} > 0.05$ and $p_{3/4} \ll 10^{-4}$. 
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- **Key problem:** How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed uniformly.
- Which lattice is optimal? The hexagonal lattice
- **Q1:** How big should the hexagons be?
- **Q2:** Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- Key problem: How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed uniformly.
- Which lattice is optimal? The hexagonal lattice
- Q1: How big should the hexagons be?
- Q2: Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- Key problem: How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed uniformly.
- Which lattice is optimal? The hexagonal lattice
  Q1: How big should the hexagons be?
- Q2: Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- **Key problem:** How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed **uniformly**.
- Which lattice is optimal? **The hexagonal lattice**
  - Q1: How big should the hexagons be?
  - Q2: Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- **Key problem:** How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed **uniformly**.
- Which lattice is optimal? The hexagonal lattice
  Q1: How big should the hexagons be?
- Q2: Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- **Key problem:** How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed uniformly.
- Which lattice is optimal? The **hexagonal lattice**
  - **Q1:** How big should the hexagons be?
  - **Q2:** Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

Focus on 2-d (results generalize to higher dimensions)

Sources = hospitals, post offices, pubs, ... 

Key problem: How do we cope with uneven population densities?

Obvious: if density is uniform then sources are best distributed uniformly.

Q1: Which lattice is optimal? The hexagonal lattice

Q2: How big should the hexagons be?

Many sources, many sinks

Q1: How big should the hexagons be?

Q2: Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- **Key problem:** How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed **uniformly**.
- Which lattice is optimal? The **hexagonal lattice**
  - **Q1:** How big should the hexagons be?
  - **Q2:** Given population density is uneven, what do we do?
Many sources, many sinks

How do we distribute sources?

- Focus on 2-d (results generalize to higher dimensions)
- Sources = hospitals, post offices, pubs, ...
- Key problem: How do we cope with uneven population densities?
- Obvious: if density is uniform then sources are best distributed uniformly.
- Which lattice is optimal? The hexagonal lattice
  Q1: How big should the hexagons be?
- Q2: Given population density is uneven, what do we do?
Solidifying the basic problem

- Given a region with some population distribution $\rho$, most likely uneven.
- Given resources to build and maintain $N$ facilities.
- **Q:** How do we locate these $N$ facilities so as to minimize the average distance between an individual’s residence and the nearest facility?
- Problem of interested and studied by geographers, sociologists, computer scientists, mathematicians, ...
Optimal source allocation

Solidifying the basic problem

- Given a region with some population distribution $\rho$, most likely uneven.
- Given resources to build and maintain $N$ facilities.
- Q: How do we locate these $N$ facilities so as to minimize the average distance between an individual’s residence and the nearest facility?
- Problem of interested and studied by geographers, sociologists, computer scientists, mathematicians, ...
- See work by Stephan $^{[19, 20]}$ and by Gastner and Newman (2006) $^{[8]}$ and work cited by them.
Optimal source allocation

Solidifying the basic problem

► Given a region with some population distribution $\rho$, most likely uneven.
► Given resources to build and maintain $N$ facilities.
► Q: How do we locate these $N$ facilities so as to minimize the average distance between an individual’s residence and the nearest facility?
► Problem of interested and studied by geographers, sociologists, computer scientists, mathematicians, ...
► See work by Stephan $^{[19, 20]}$ and by Gastner and Newman (2006) $^{[8]}$ and work cited by them.
Solidifying the basic problem

- Given a region with some population distribution $\rho$, most likely uneven.
- Given resources to build and maintain $N$ facilities.
- **Q:** How do we locate these $N$ facilities so as to minimize the average distance between an individual’s residence and the nearest facility?
- Problem of interested and studied by geographers, sociologists, computer scientists, mathematicians, ...
Optimal source allocation

Solidifying the basic problem

- Given a region with some population distribution $\rho$, most likely uneven.
- Given resources to build and maintain $N$ facilities.
- **Q:** How do we locate these $N$ facilities so as to minimize the average distance between an individual’s residence and the nearest facility?
- Problem of interested and studied by geographers, sociologists, computer scientists, mathematicians, ...
- See work by Stephan $^{[19, 20]}$ and by Gastner and Newman (2006) $^{[8]}$ and work cited by them.
Optimal source allocation

Gastner and Newman (2006)\textsuperscript{[8]}

- Approximately optimal location of 5000 facilities.
- Based on 2000 Census data.
- Simulated annealing + Voronoi tessellation.
Optimal source allocation

From Gastner and Newman (2006) [8]

- Optimal facility density $D$ vs. population density $\rho$.
  - Fit is $D \propto \rho^{0.66}$ with $r^2 = 0.94$.
  - Looking good for a 2/3 power...
Optimal source allocation

From Gastner and Newman (2006) [8]

- Optimal facility density $D$ vs. population density $\rho$.
- Fit is $D \propto \rho^{0.66}$ with $r^2 = 0.94$.
- Looking good for a $2/3$ power...
Optimal source allocation

From Gastner and Newman (2006) [8]

- Optimal facility density $D$ vs. population density $\rho$.
- Fit is $D \propto \rho^{0.66}$ with $r^2 = 0.94$.
- Looking good for a $2/3$ power...
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Optimal source allocation

Size-density law:

▶ $D \propto \rho^{2/3}$

▶ In $d$ dimensions:

$D \propto \rho^{d/(d+1)}$

▶ Why?

▶ Very different story to branching networks where there is either one source or one sink.

▶ Now sources & sinks are distributed throughout region...
Optimal source allocation

Size-density law:

\[ D \propto \rho^{2/3} \]

In \( d \) dimensions:

\[ D \propto \rho^{d/(d+1)} \]

Why?

Very different story to branching networks where there is either one source or one sink.

Now sources & sinks are distributed throughout region...
Optimal source allocation

Size-density law:

- \( D \propto \rho^{2/3} \)

- In \( d \) dimensions:
  
  \[
  D \propto \rho^{d/(d+1)}
  \]

- Why?
  
  - Very different story to branching networks where there is either one source or one sink.
  
  - Now sources & sinks are distributed throughout region...
Optimal source allocation

Size-density law:

- $D \propto \rho^{2/3}$
- In $d$ dimensions:
  
  $$D \propto \rho^{d/(d+1)}$$

- Why?
- Very different story to branching networks where there is either one source or one sink.
- Now sources & sinks are distributed throughout region...
Optimal source allocation

Size-density law:

- $D \propto \rho^{2/3}$

- In $d$ dimensions:
  
  $D \propto \rho^{d/(d+1)}$

- Why?

- Very different story to branching networks where there is either one source or one sink.

- Now sources & sinks are distributed throughout region...
Optimal source allocation

- One treatment due to Stephan’s (1977)\[^{19, 20}\]: “Territorial Division: The Least-Time Constraint Behind the Formation of Subnational Boundaries” (Science, 1977)

- Zipf-like approach: invokes principle of minimal effort.

- Also known as the Homer principle.
Optimal source allocation

- One treatment due to Stephan’s (1977) [19, 20]: “Territorial Division: The Least-Time Constraint Behind the Formation of Subnational Boundaries” (Science, 1977)
- Zipf-like approach: invokes principle of minimal effort.
- Also known as the Homer principle.
Optimal source allocation

- One treatment due to Stephan’s (1977) \([19, 20]\): “Territorial Division: The Least-Time Constraint Behind the Formation of Subnational Boundaries” (Science, 1977)
- Zipf-like approach: invokes principle of minimal effort.
- Also known as the Homer principle.
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Size-density law

Deriving the optimal source distribution:

- Stronger result obtained by Gusein-Zade (1982). \[10\]
- Basic idea: Minimize the average distance from a random individual to the nearest facility.
- Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
- Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| d\vec{x}.$$ 

- Also known as the p-median problem.
- Not easy... in fact this one is an NP-hard problem. \[8\]
Size-density law

Deriving the optimal source distribution:

- Basic idea: Minimize the average distance from a random individual to the nearest facility.

- Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
- Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| d\vec{x}.$$ 

- Also known as the p-median problem.
- Not easy... in fact this one is an NP-hard problem. [8]
Size-density law

Deriving the optimal source distribution:

- Stronger result obtained by Gusein-Zade (1982). \[^{10}\]
- **Basic idea:** Minimize the average distance from a random individual to the nearest facility.
- Assume given a fixed population density \( \rho \) defined on a spatial region \( \Omega \).

  Formally, we want to find the locations of \( n \) sources \( \{ \vec{x}_1, \ldots, \vec{x}_n \} \) that minimizes the cost function

  \[
  F(\{ \vec{x}_1, \ldots, \vec{x}_n \}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| \, d\vec{x}.
  \]

  Also known as the p-median problem.

- Not easy... in fact this one is an NP-hard problem. \[^{8}\]
Size-density law

Deriving the optimal source distribution:

- Stronger result obtained by Gusein-Zade (1982). \[10\]
- **Basic idea:** Minimize the average distance from a random individual to the nearest facility.
- Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
- Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| \, d\vec{x}.$$ 

- Also known as the p-median problem.
- Not easy... in fact this one is an NP-hard problem. \[8\]
Size-density law

Deriving the optimal source distribution:

- **Basic idea:** Minimize the average distance from a random individual to the nearest facility.
- Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
- Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_{i} ||\vec{x} - \vec{x}_i|| \, d\vec{x}.$$  

- Also known as the p-median problem.
- Not easy... in fact this one is an NP-hard problem. [8]
Size-density law

Deriving the optimal source distribution:

- **Basic idea:** Minimize the average distance from a random individual to the nearest facility.
- Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
- Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| \, d\vec{x}.$$ 

- Also known as the p-median problem.
- Not easy... in fact this one is an NP-hard problem. [8]
Size-density law

Deriving the optimal source distribution:

▶ Basic idea: Minimize the average distance from a random individual to the nearest facility.
▶ Assume given a fixed population density $\rho$ defined on a spatial region $\Omega$.
▶ Formally, we want to find the locations of $n$ sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

$$F(\{\vec{x}_1, \ldots, \vec{x}_n\}) = \int_{\Omega} \rho(\vec{x}) \min_i ||\vec{x} - \vec{x}_i|| d\vec{x}.$$ 

▶ Also known as the p-median problem.
▶ Not easy... in fact this one is an NP-hard problem. [8]
Size-density law

Can (roughly) turn into a Lagrange multiplier story:

- By varying \( \{\vec{x}_1, ..., \vec{x}_n\} \), minimize

\[
G(A) = c \int_{\Omega} \rho(\vec{x}) A(\vec{x})^{1/2} d\vec{x} - \lambda \left( n - \int_{\Omega} [A(\vec{x})]^{-1} d\vec{x} \right)
\]

- Involves estimating typical distance from \( \vec{x} \) to the nearest source (say \( i \)) as \( c_i A(\vec{x})^{1/2} \) where \( c_i \) is a shape factor for the \( i \)th Voronoi cell.

- **Sneakiness:** set \( c_i = c \).

- Compute \( \delta G/\delta A \), the functional derivative (\( \Box \)).

- Solve and substitute \( D = 1/A \), we find

\[
D(\vec{x}) = \left( \frac{c}{2\lambda \rho} \right)^{2/3}.
\]
Size-density law

Can (roughly) turn into a Lagrange multiplier story:

- By varying \( \{\vec{x}_1, \ldots, \vec{x}_n\} \), minimize

\[
G(A) = c \int_{\Omega} \rho(\vec{x}) A(\vec{x})^{1/2} \, d\vec{x} - \lambda \left( n - \int_{\Omega} [A(\vec{x})]^{-1} \, d\vec{x} \right)
\]

- Involves estimating typical distance from \( \vec{x} \) to the nearest source (say \( i \)) as \( c_i A(\vec{x})^{1/2} \) where \( c_i \) is a shape factor for the \( i \)th Voronoi cell.

- **Sneakiness:** set \( c_i = c \).

- Compute \( \delta G / \delta A \), the functional derivative (\( \delta \)).

- Solve and substitute \( D = 1 / A \), we find

\[
D(\vec{x}) = \left( \frac{c}{2\lambda \rho} \right)^{2/3}
\]
Size-density law

Can (roughly) turn into a Lagrange multiplier story:

- By varying \( \{\vec{x}_1, ..., \vec{x}_n\} \), minimize

\[
G(A) = c \int_{\Omega} \rho(\vec{x}) A(\vec{x})^{1/2} d\vec{x} - \lambda \left(n - \int_{\Omega} [A(\vec{x})]^{-1} d\vec{x}\right)
\]

- Involves estimating typical distance from \( \vec{x} \) to the nearest source (say \( i \)) as \( c_i A(\vec{x})^{1/2} \) where \( c_i \) is a shape factor for the \( i \)th Voronoi cell.

- **Sneakiness:** set \( c_i = c \).

- Compute \( \frac{\delta G}{\delta A} \), the functional derivative.

- Solve and substitute \( D = 1/A \), we find

\[
D(\vec{x}) = \left(\frac{c}{2\lambda \rho}\right)^{2/3}.
\]
Size-density law

Can (roughly) turn into a Lagrange multiplier story:

- By varying $\{\vec{x}_1, \ldots, \vec{x}_n\}$, minimize

$$G(A) = c \int_{\Omega} \rho(\vec{x}) A(\vec{x})^{1/2} d\vec{x} - \lambda \left( n - \int_{\Omega} [A(\vec{x})]^{-1} d\vec{x} \right)$$

- Involves estimating typical distance from $\vec{x}$ to the nearest source (say $i$) as $c_i A(\vec{x})^{1/2}$ where $c_i$ is a shape factor for the $i$th Voronoi cell.

- **Sneakiness:** set $c_i = c$.

- Compute $\delta G/\delta A$, the **functional derivative** ($\boxdot$).

- Solve and substitute $D = 1/A$, we find

$$D(\vec{x}) = \left( \frac{c}{2\lambda \rho} \right)^{2/3}.$$
Size-density law

Can (roughly) turn into a Lagrange multiplier story:

- By varying \{\vec{x}_1, ..., \vec{x}_n\}, minimize

\[
G(A) = c \int_\Omega \rho(\vec{x}) A(\vec{x})^{1/2} d\vec{x} - \lambda \left( n - \int_\Omega [A(\vec{x})]^{-1} d\vec{x} \right)
\]

- Involves estimating typical distance from \vec{x} to the nearest source (say \(i\)) as \(c_i A(\vec{x})^{1/2}\) where \(c_i\) is a shape factor for the \(i\)th Voronoi cell.

- **Sneakiness:** set \(c_i = c\).

- Compute \(\delta G/\delta A\), the functional derivative ( المباشر).

- Solve and substitute \(D = 1/A\), we find

\[
D(\vec{x}) = \left( \frac{c}{2\lambda \rho} \right)^{2/3}
\]
Outline

Single Source

Distributed Sources
   Facility location
   Size-density law
   A reasonable derivation
   Global redistribution networks

Structure Detection
   Hierarchy by division
   Hierarchy by shuffling
   Spectral methods
   Hierarchies & Missing Links
   General structure detection

Final words

References
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:
  \[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

  Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:
  \[(1 - \delta)\ell_{ij} + \delta(\#\text{hops}).\]

  When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:

\[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

- Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:

\[ (1 - \delta)\ell_{ij} + \delta(\#\text{hops}). \]

- When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:

  \[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

- Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:

  \[ (1 - \delta)\ell_{ij} + \delta(\# \text{hops}). \]

- When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:

\[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

- Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:

\[ (1 - \delta) \ell_{ij} + \delta(\#\text{hops}). \]

- When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:

\[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

- Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:

\[ (1 - \delta)\ell_{ij} + \delta(\#\text{hops}). \]

- When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

One more thing:

- How do we supply these facilities?
- How do we best redistribute mail? People?
- How do we get beer to the pubs?
- Gaster and Newman model: cost is a function of basic maintenance and travel time:

\[ C_{\text{maint}} + \gamma C_{\text{travel}}. \]

- Travel time is more complicated: Take ‘distance’ between nodes to be a composite of shortest path distance \( \ell_{ij} \) and number of legs to journey:

\[ (1 - \delta)\ell_{ij} + \delta(\#\text{hops}). \]

- When \( \delta = 1 \), only number of hops matters.
Global redistribution networks

From Gastner and Newman (2006) [8]
Structure detection

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

Zachary’s karate club [25, 16]
We now turn to applications of our methods to real-world network data. Our first such example is taken from one of the classic studies in social network analysis. Over the course of two years in the early 1970s, Wayne Zachary observed social interactions between the members of a karate club at an American university.

Zachary's karate club network

The network as presented in Fig. 10 was constructed by Thomas Farley, who collected the raw social data over the course of the study. He then constructed networks of ties across many scales?

The shapes of the networks vary in proportion to the number of pairs of scientists who have collaborated. Now the overall structure of the network is centered around the administrator after the fission of the club, open circles those who raised club fees, and as a result the club eventually split in two, forming two smaller clubs, centered around the administrator and the other, and collaboration on any topic is a reasonable indication of acquaintance.

In Fig. 8, we show a consensus network structure extracted from the Zachary social network. The network is centered around the administrator after the fission of the club, open circles those who raised club fees, and as a result the club eventually split in two, forming two smaller clubs, centered around the administrator and the other, and collaboration on any topic is a reasonable indication of acquaintance.

In Fig. 9, we show the dendrograms for a hierarchical clustering of the Zachary social network. In the leftmost two panels, we show the dendrograms for the shortest-path version of our algorithms and the random-walk version of our algorithms. In the next two panels, we show the dendrograms for the shortest-path version of our algorithms and the random-walk version of our algorithms. In the next two panels, we show the dendrograms for the shortest-path version of our algorithms and the random-walk version of our algorithms. In the next two panels, we show the dendrograms for the shortest-path version of our algorithms and the random-walk version of our algorithms. In the last panel of Fig. 9, we show the dendrogram and the corresponding division of the Zachary social network into 5 communities—around 0.4 in each case—indicating that there are many divisions both within the club and outside it. By chance, a division suggested by which group each club administrator aligned with, after the fission of the club, open circles those who raised club fees, and as a result the club eventually split in two, forming two smaller clubs, centered around the administrator and the other, and collaboration on any topic is a reasonable indication of acquaintance.
Structure detection

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

Zachary’s karate club [25, 16]

Possible substructures:
hierarchies, cliques, rings, …

Plus:
All combinations of substructures.

Much focus on hierarchies…
Structure detection

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

Zachary’s karate club [25, 16]

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

Plus:
All combinations of substructures.

Much focus on hierarchies...
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.


- See also
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.

- Following comes from “Finding and evaluating community structure in networks” by Newman and Girvan (PRE, 2004).[^16]

- See also
  2. “Community structure in social and biological networks” by Girvan and Newman (PNAS, 2002).[^9]
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.


- See also
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.
- See also
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.
- See also
**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.
Hierarchy by division

Key element:

- 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:

\[
Q = \sum_i \left[ e_{ii} - (\sum_j e_{ij})^2 \right] = \text{Tr} \cdot E - ||E^2||_1,
\]
where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
Hierarchy by division

Key element:

- 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:

\[ Q = \sum |e_{ii} - (\sum e_{ij})^2| = 2 \cdot \text{Tr} \cdot E - ||E^2||, \]

where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
Hierarchy by division

Key element:

- 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:

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

where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
Hierarchy by division

Key element:

- 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:

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

where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
Hierarchy by division

Key element:

- 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:

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

where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
Hierarchy by division

Key element:

- 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:

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

where \( e_{ij} \) is the fraction of edges between identified communities \( i \) and \( j \).
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:
  \[
  \langle k \rangle_{in} = 6 \text{ and } \langle k \rangle_{out} = 2.
  \]
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

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

Test case:

- Generate random community-based networks.
- $N = 128$ with four communities of size 32.
- Add edges randomly within and across communities.
- Example:
  
  \[
  \langle k \rangle_{\text{in}} = 6 \quad \text{and} \quad \langle k \rangle_{\text{out}} = 2.
  \]
Maximum modularity $Q \approx 0.5$ obtained when four communities are uncovered.

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

Further ‘discovery’ of internal structure is somewhat meaningless, as any communities arise accidentally.
Factions in Zachary’s karate club network. [25]
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 \(\ell, |I_{\ell, st}|\).

- Sum \(|I_{\ell, st}|\) over all pairs of nodes to obtain electronic betweenness for edge \(\ell\).
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes \(i\) and \(j\):

\[
B_{ij}^{\text{elec}} = a_{ij}|V_i - V_j|.
\]

- Upshot: specific measure of betweenness not too important.
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 $\ell$, $|I_{\ell, st}|$.
- Sum $|I_{\ell, st}|$ over all pairs of nodes to obtain electronic betweenness for edge $\ell$.
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes $i$ and $j$:
  \[ B_{ij}^{\text{elec}} = a_{ij} |V_i - V_j|. \]
- Upshot: specific measure of betweenness not too important.
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 $\ell$, $|I_{\ell,st}|$.

- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge $\ell$.
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes $i$ and $j$: 
  \[ B_{ij}^{\text{elec}} = a_{ij}|V_i - V_j|. \]

- **Upshot:** specific measure of betweenness not too important.
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 $\ell$, $|I_{\ell,st}|$.

- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain **electronic betweenness** for edge $\ell$.
  - (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes $i$ and $j$:
  \[ B_{ij}^{\text{elec}} = a_{ij} |V_i - V_j|. \]

- **Upshot**: specific measure of betweenness not too important.
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 $\ell$, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge $\ell$.
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes $i$ and $j$:

\[ B_{ij}^{elec} = a_{ij}|V_i - V_j|. \]

- Upshot: specific measure of betweenness not too important.
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 \( \ell \), \( |I_{\ell,st}| \).
- Sum \( |I_{\ell,st}| \) over all pairs of nodes to obtain electronic betweenness for edge \( \ell \).
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes \( i \) and \( j \):

\[
B_{ij}^{\text{elec}} = a_{ij} |V_i - V_j|.
\]

- Upshot: specific measure of betweenness not too important.
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 $\ell$, $|I_{\ell,st}|$.

- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge $\ell$.
- (Equivalent to random walk betweenness.)
- Electronic betweenness for edge between nodes $i$ and $j$:
  
  \[ B_{ij}^{\text{elec}} = a_{ij} |V_i - V_j|. \]

- Upshot: specific measure of betweenness not too important.
Scientists working on networks

FIG. 10. Illustration of the use of the community-structure algorithm to make sense of a complex network.

The initial network is a network of coauthorships between physicists who have published on topics related to networks. The figure shows only the largest component of the network, which contains 145 scientists. There are 90 more scientists in smaller components, which are not shown.

Application of the shortest-path betweenness version of the community-structure algorithm produces the communities indicated by the shades of the vertices.

A coarse-graining of the network in which each community is represented by a single node, with edges representing collaborations between communities. The thickness of the edges is proportional to the number of pairs of collaborators between communities. Clearly panel reveals much that is not easily seen in the original network of panel a.
Scientists working on networks
Dolphins!

The split into two groups appears to correspond to a known division of the dolphin community. Lusseau reports that for a period of about two years during observation of the dolphins they separated into two groups along the lines found by our analysis, apparently because of the disappearance of individuals on the boundary between the groups. When some of these individuals later reappeared, the two halves of the network joined together once more. As Lusseau points out, developments of this kind illustrate that the dolphin network is not merely a scientific curiosity but, like human social networks, is closely tied to the evolution of the community. The subgroupings within the larger half of the network also seem to correspond to real divisions among the animals: the largest subgroup consists almost entirely of females and the others almost entirely of males, and it is conjectured that the split between the male groups is governed by matrilineage.

Figure 12 shows the community structure of the network of interactions between major characters in Victor Hugo’s sprawling novel of crime and redemption in post-restoration France, Les Misérables. The network has been drawn with longer edges between vertices in different communities than between those in the same community, to make the community groupings clearer. The same is also true of Figs. 12 and 13.
The split into two groups appears to correspond to a known division of the dolphin community. Lusseau reports that for a period of about two years during observation of the dolphins they separated into two groups along the lines found by our analysis, apparently because of the disappearance of individuals on the boundary between the groups. When some of these individuals later reappeared, the two halves of the network joined together once more. As Lusseau points out, developments of this kind illustrate that the dolphin network is not merely a scientific curiosity but, like human social networks, is closely tied to the evolution of the community. The subgroupings within the larger half of the network also seem to correspond to real divisions among the animals: the largest subgroup consists almost entirely of females and the others almost entirely of males, and it is conjectured that the split between the male groups is governed by matrilineage.

Figure 12 shows the community structure of the network of interactions between major characters in Victor Hugo's sprawling novel of crime and redemption in post-restoration France. The greatest modularity achieved in the shortest-path version of our algorithm is $Q = 0.54$ and corresponds to the 11 communities shown.
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Shuffling for structure

▶ “Extracting the hierarchical organization of complex systems”
Sales-Pardo et al., PNAS (2007) [17, 18]

▶ Consider all partitions of networks into \( m \) groups

▶ As for Newman and Girvan approach, aim is to find partitions with maximum modularity:

\[
Q = \sum_i \left[ e_{ii} - \left( \sum_j e_{ij} \right)^2 \right] = \text{Tr}E - \|E^2\|_1.
\]
Shuffling for structure

- “Extracting the hierarchical organization of complex systems”
  Sales-Pardo et al., PNAS (2007) [17, 18]
- 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) [17, 18]
- 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

- 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 $A_{ij}$.
- $A_{ij} = \Pr$ 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 $A_{ij}$.
- $A_{ij} = \text{Pr}$ 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 $A_{ij}$.
- $A_{ij} = \Pr$ 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 $j = \text{# matching neighbors for } i \text{ and } j \text{ divided by maximum of } k_i \text{ and } k_j$. 
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 $A_{ij}$.
  - $A_{ij} = \Pr$ 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 $A_{ij}$.
- $A_{ij} = \Pr$ 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 $A_{ij}$.

\[ A_{ij} = \Pr \text{ random walker on modularity network ends up at a partition with } i \text{ and } j \text{ in the same group.} \]

- C.f. **topological overlap** between $i$ and $j$ = # matching neighbors for $i$ and $j$ divided by maximum of $k_i$ and $k_j$. 
A: Base network; B: Partition network; C: Coclassification matrix; D: Comparison to random networks (all the same!); E: Ordered coclassification matrix;
A: Base network; B: Partition network; C: Coclassification matrix; D: Comparison to random networks (all the same!); E: Ordered coclassification matrix; Conclusion: no structure...
Shuffling for 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

\[ C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} |i - j|. \]

- Use simulated annealing (slow).
Shuffling for 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

\[ C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} |i - j|. \]

- Use simulated annealing (slow).
Shuffling for 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

\[ C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} |i - j|. \]

- Use simulated annealing (slow).
Shuffling for 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
    \[
    C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} |i - j|.
    \]
    - Use simulated annealing (slow).
Shuffling for 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

\[ C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} |i - j|. \]

- Use simulated annealing (slow).
Shuffling for 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

\[
C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij}|i - j|.
\]

- Use simulated annealing (slow).
Let then \( P_{\text{max}} \) be the set of partitions for which the modularity \( M \) is a local maxima, that is, partitions for which neither the change of a single node transforms one partition into the other. This landscape has a very clear modular structure. However, such a procedure would not take into consideration the hierarchical organization of the nodes. We show the affinity matrices obtained for two different measures: (A) logical overlap (see SI Text 640) and (B) box-model clustering. The box-model clustering and the "box clustering" method we propose are sampled with a very clear modular structure (see Fig. 2). Note that topological overlap fails to find any hierarchical structure (see Fig. 2). In contrast, the box-model classification clearly detects the three-level hierarchical structure that the affinity matrix displays, whereas the hierarchical tree obtained via hierarchical clustering fails to reproduce the clear three-level hierarchical pattern. In the hierarchical clustering tree, the vertical axis shows the average distance, \( d \). At the far right, we show the hierarchical tree determined with the "nested-box" pattern along the diagonal. Significantly, the hierarchical tree obtained via hierarchical clustering and the "box clustering" method we propose is as good at detecting communities as a standard community detection algorithm for networks with a flat organization of the nodes. Additionally, our method is able to detect the top level for all cases analyzed, whereas standard modularity optimization algorithms are not.

- \( N = 640 \),
- \( \langle k \rangle = 16 \),
- 3 tiered hierarchy.
Air transportation:

- Modules found match up with geopolitical units.
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
General structure detection


▶ Consider normal matrix $K^{-1}A$, random walk matrix $A^TK^{-1}$, Laplacian $K - A$, and $AA^T$.

▶ Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.

▶ Build on Kleinberg’s HITS algorithm. [13]
General structure detection

- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Build on Kleinberg’s HITS algorithm. [13]
General structure detection

- Consider normal matrix $\mathbf{K}^{-1}\mathbf{A}$, random walk matrix $\mathbf{A}^T\mathbf{K}^{-1}$, Laplacian $\mathbf{K} - \mathbf{A}$, and $\mathbf{AA}^T$.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Build on Kleinberg’s HITS algorithm. [13]
General structure detection

▶ “Detecting communities in large networks” Capocci et al. (2005) \[4\]

▶ Consider normal matrix $K^{-1}A$, random walk matrix $A^TK^{-1}$, Laplacian $K - A$, and $AA^T$.

▶ Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.

▶ Build on Kleinberg’s HITS algorithm. \[13\]
General structure detection

Example network:

Thus, solving the eigenproblem is equivalent to minimizing function (1) with constraint (2), where the $x_i$'s are eigenvectors components. The absolute minimum corresponds to the trivial eigenvector, which is constant. The other stationary points correspond to eigenvectors where components associated to well connected nodes assume similar values.

In order to compute cluster sizes and distribution, methods such as bisection or edge-betweenness based ones are very poor in detecting the end of the recursive splitting. Our approach, instead, immediately detects the number of clear clusters from the eigenvectors profile.

As an illustrative example, we show in Fig. 2 the profile of the second eigenvectors of $D/C_0W$ corresponding to the simple graph shown in Fig. 1 with $S = 19$ nodes, where random weights between 1 and 10 were assigned to the links ($Figs. 1 and 2$).

The components of the eigenvectors assume approximately constant values on nodes belonging to the same community. Thus, the number of communities emerges naturally and it is not needed as input.

However, as aforementioned, when dealing with large networks with no clear partitioning, the precise value of the eigenvector components is of little use. In such situations, the typical eigenvector profile is not step-like, but resembles a continuous curve. Nevertheless, our method can still be applied, and efficiently detects sets of well connected nodes. In fact, components corresponding to nodes belonging to the same communities are still strongly correlated taking, in each eigenvector, similar values among themselves. Thus, a natural way to identify communities in an automatic manner, is to measure the correlation $r_{ij} = \frac{\langle x_i x_j \rangle}{\sqrt{\langle x_i^2 \rangle \langle x_j^2 \rangle}}$, (4), where the average $\langle \rangle$ is over the first few nontrivial eigenvectors. The quantity $r_{ij}$ measures the community closeness between nodes $i$ and $j$.

Though the performance may be improved by averaging over more and more eigenvectors, with increased computational effort, we find that indeed a small number of eigenvectors suffices to identify the community to which nodes belong, even in large networks.
When dealing with a directed network, links do not correspond to any equivalence relation. Rather, pointing to common neighbors is a significant relation, as suggested in the sociologists' literature where this quantity measures the so-called structural equivalence of nodes [18]. Accordingly, in a directed network, clusters should be composed by nodes pointing to a high number of common neighbors, no matter their direct linkage. For directed networks, we thus modify our method in the streamline of the HITS algorithm [17]. The HITS algorithm was proposed on empirical bases to find the main communities in large oriented networks. It assumes that the largest components (in the absolute value) of eigenvectors of the matrices $A A^T$ and $A^T A$ correspond to highly clustered nodes belonging to a single community. Such algorithm efficiently detects the main communities, even when these are not sharply defined. However, it becomes computationally heavy when one is interested in minor communities, which correspond to smaller eigenvalues. As explained in the undirected case, we tackle this issue by combining information from the first few eigenvectors of the normal matrix and extracting the community structure from correlations between the same components in different eigenvectors.

To detect the community structure in a directed network, we therefore replace, in the previous analysis, the matrix $W$ with a matrix $Y = W W^T$. This corresponds to replacing the directed network with an undirected weighted network, where nodes pointing to common neighbors are connected by a link, whose intensity is proportional to the total sum of the weights of the links pointing from the two original nodes to the common neighbors. Then, one performs the analysis on the undirected network as described previously. Thus, the function to minimize in this case is

$$y(x) = \sum_{i,j,l} S_{ijl}(x_i/C_0 x_j)^2 w_{il} w_{jl}. \ (5)$$
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
  General structure detection

Final words

References
Hierarchies and missing links

- 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)\textsuperscript{[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)\textsuperscript{[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

- 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>
<thead>
<tr>
<th>Table 1</th>
<th>Comparison of original and resampled networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>$\langle k \rangle_{real}$</td>
</tr>
<tr>
<td>T. pallidum</td>
<td>4.8</td>
</tr>
<tr>
<td>Terrorists</td>
<td>4.9</td>
</tr>
<tr>
<td>Grassland</td>
<td>3.0</td>
</tr>
</tbody>
</table>

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 dendrogram for grassland species.
- Copes with disassortative and assortative communities.
Hierarchies and missing links

- **Consensus dendrogram** for grassland species.
- Copes with disassortative and assortative communities.
Outline

Single Source

Distributed Sources
  Facility location
  Size-density law
  A reasonable derivation
  Global redistribution networks

Structure Detection
  Hierarchy by division
  Hierarchy by shuffling
  Spectral methods
  Hierarchies & Missing Links
    General structure detection

Final words

References
General structure detection

“The discovery of structural form”
Kemp and Tenenbaum, PNAS (2008)[12]
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.
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

Performance for test networks.

True Partition Chain Ring Tree Grid

Fig. 1: The hierarchical generative model can be used for many purposes. If the form of a data set is already known, we can search for the best structure by enumerating over possible forms and selecting the one with the highest log probability. If the form of the data is not known, at least two strategies might be tried. For some applications it may be desirable to integrate over the space of structures, computing the posterior probabilities of each form. Our method for identifying the best structures found is to search for the best structures in each form in a blind fashion, attempting to split each cluster node several times. Kemp and Tenenbaum (2008) show that the chain-structured data is most likely to be true of pairs (x, y) such that there is an edge between cluster S = \( \alpha \) and \( |D \beta \) = \( \alpha \). As before, the terms in the graph are added to the log probabilities along each y axis so that the worst performance for test networks.

Frame 67/78
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
Effect of adding features on detected form.

- **Straight partition**
  - simple tree
  - complex tree
Final words:

Science in three steps:

1. Find interesting/meaningful/important phenomena involving spectacular amounts of data.
2. Describe what you see.
3. Explain it.

A plea/warning

Beware your assumptions—don’t use tools/models because they’re there, or because everyone else does...
Final words:

Science in three steps:

1. Find interesting/meaningful/important phenomena involving spectacular amounts of data.
2. Describe what you see.
3. Explain it.

A plea/warning

Beware your assumptions—don’t use tools/models because they’re there, or because everyone else does...
Final words:

Science in three steps:

1. Find interesting/meaningful/important phenomena involving spectacular amounts of data.
2. Describe what you see.
3. Explain it.

A plea/warning

Beware your assumptions—don’t use tools/models because they’re there, or because everyone else does...
Final words:

Science in three steps:

1. Find interesting/meaningful/important phenomena involving spectacular amounts of data.
2. Describe what you see.
3. Explain it.

A plea/warning

Beware your assumptions—don’t use tools/models because they’re there, or because everyone else does...
More final words:

A real theory of everything:

1. Is not just about the small stuff...
2. It’s about the increase of complexity
More final words:

A real theory of everything:

1. Is not just about the small stuff...
2. It’s about the increase of complexity
More final words:

A real theory of everything:

1. Is not just about the small stuff...
2. It’s about the increase of complexity
More final words:

A real theory of everything:

1. Is not just about the small stuff...
2. It’s about the increase of complexity

Symmetry breaking/ Accidents of history vs. Universality
More final words:

A real theory of everything:

1. Is not just about the small stuff...
2. It’s about the increase of complexity

Symmetry breaking/ Accidents of history vs. Universality

How probable is a certain level of complexity?
References I

Topology of the fittest transportation network. 

Size and form in efficient transportation networks. 

Structure, scaling, and phase transition in the optimal transport network. 
References II

Detecting communities in large networks.
[PDF](http://example.com)

Hierarchical structure and the prediction of missing links in networks.
[PDF](http://example.com)

Unified view of scaling laws for river networks.
[PDF](http://example.com)
References III


References IV

Size and power in mammals.

The discovery of structural form.
[link to PDF]

Authoritative sources in a hyperlinked environment.
[link to PDF]

Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.
[link to PDF]
References V


References VI


References VII


