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

Outline

Optimal supply networks

What's the best way to distribute stuff?

- Stuff = medical services, energy, nutrients, people, ...
- Some fundamental network problems:
    1. Distribut e 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. Distribut e 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. Distribut e 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. Distribut e 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. Distribut e 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. Distribut e 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. Distribut e 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. Distribut e 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
  and
  Z_j = link j's impedance?
- Example: $\gamma = 2$ for electrical networks.

Single source optimal supply

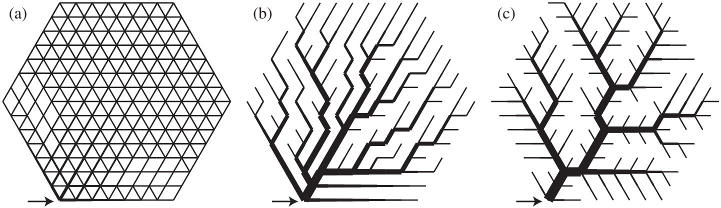
Basic Q for distribution/supply networks:

- How does flow behave given cost:

$$C = \sum_j I_j^{\gamma} Z_j$$

  where
  I_j = current on link j
  and
  Z_j = link j's impedance?
- Example: $\gamma = 2$ for electrical networks.

Single source optimal supply



(a) $\gamma > 1$: Braided (bulk) flow

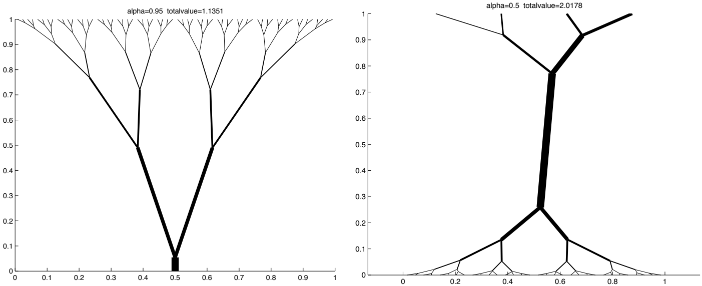
(b) $\gamma < 1$: Local minimum: Branching flow

(c) $\gamma < 1$: Global minimum: Branching flow

From Bohn and Magnasco [3]
See also Banavar et al. [1]

Single source optimal supply

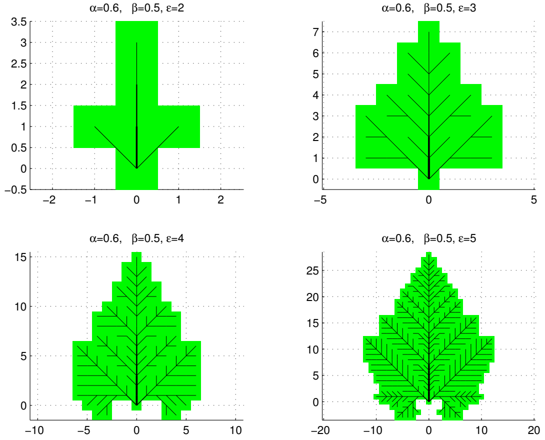
Optimal paths related to transport (Monge) problems:



Xia (2003) [24]

Growing networks:

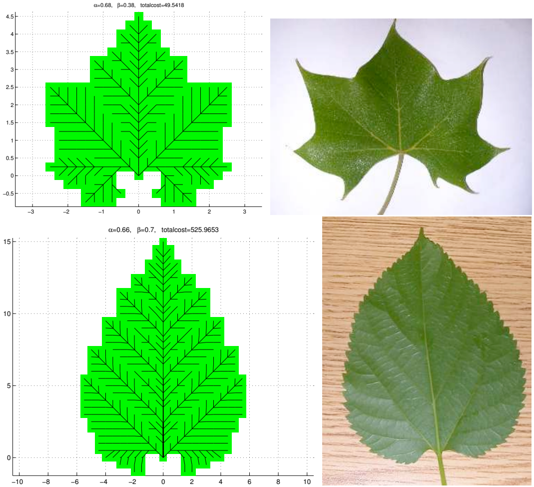
FIGURE 1. $\alpha = 0.6, \beta = 0.5$



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.

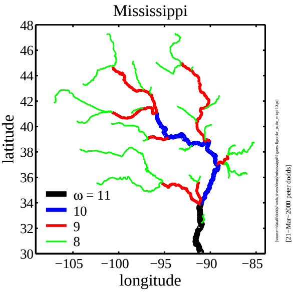
Stream Ordering:

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

- If streams of different orders ω_1 and ω_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 δ is the Kronecker delta.



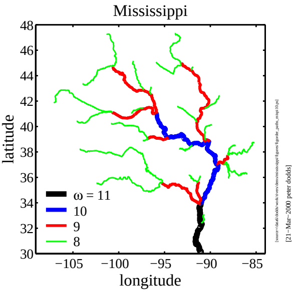
Stream Ordering:

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

- If streams of different orders ω_1 and ω_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 δ is the Kronecker delta.

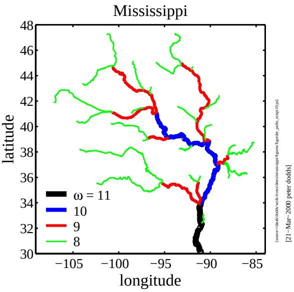


Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders ω_1 and ω_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 δ is the Kronecker delta.

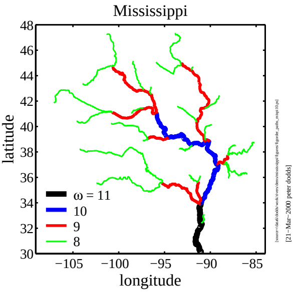


Stream Ordering:

- Label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
- Whenever two streams of the same order (ω) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders ω_1 and ω_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 δ is the Kronecker delta.



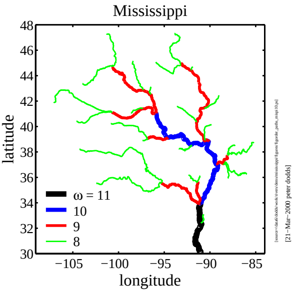
Stream Ordering:

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

- If streams of different orders ω_1 and ω_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 δ is the Kronecker delta.



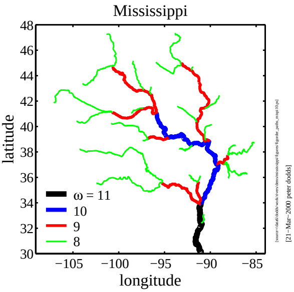
Stream Ordering:

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

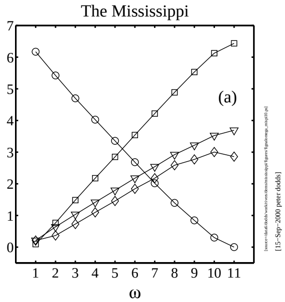
- If streams of different orders ω_1 and ω_2 meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:

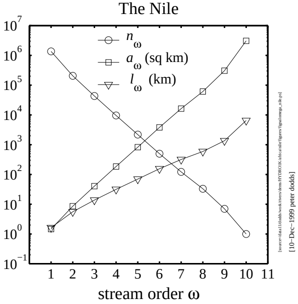
$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1,\omega_2}$$

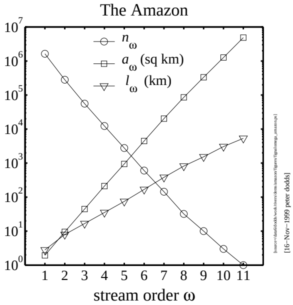
where δ is the Kronecker delta.



Horton’s laws in the real world:







Many scaling laws, many connections

relation:	scaling relation/parameter: [6]
$\ell \sim L^d$	d
$T_k = T_1 (R_T)^{k-1}$	$T_1 = R_n - R_s - 2 + 2R_s/R_n$
	$R_T = R_s$
$n_\omega / n_{\omega+1} = R_n$	R_n
$\bar{a}_{\omega+1} / \bar{a}_\omega = R_a$	$R_a = R_n$
$\bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell$	$R_\ell = R_s$
$\ell \sim a^h$	$h = \log R_s / \log R_n$
$a \sim L^D$	$D = d/h$
$L_\perp \sim L^H$	$H = d/h - 1$
$P(a) \sim a^{-\tau}$	$\tau = 2 - h$
$P(\ell) \sim \ell^{-\gamma}$	$\gamma = 1/h$
$\Lambda \sim a^\beta$	$\beta = 1 + h$
$\lambda \sim L^\varphi$	$\varphi = d$

Only 3 parameters are independent... [6]

Reported parameter values: [6]

Parameter:	Real networks:
R_n	3.0–5.0
R_a	3.0–6.0
$R_\ell = R_T$	1.5–3.0
T_1	1.0–1.5
d	1.1 ± 0.01
D	1.8 ± 0.1
h	0.50–0.70
τ	1.43 ± 0.05
γ	1.8 ± 0.1
H	0.75–0.80
β	0.50–0.70
φ	1.05 ± 0.05

Data from real blood networks

Network	R_n	R_r^{-1}	R_ℓ^{-1}	$-\frac{\ln R_r}{\ln R_n}$	$-\frac{\ln R_\ell}{\ln R_n}$	α
West *et al.*	–	–	–	0.5	$0.3\bar{3}$	0.75
rat (PAT)	2.76	1.58	1.60	0.45	0.46	0.73
cat (PAT) (Turcotte *et al.*[21])	3.67	1.71	1.78	0.41	0.44	0.79
dog (PAT)	3.69	1.67	1.52	0.39	0.32	0.90
pig (LCX)	3.57	1.89	2.20	0.50	0.62	0.62
pig (RCA)	3.50	1.81	2.12	0.47	0.60	0.65
pig (LAD)	3.51	1.84	2.02	0.49	0.56	0.65
human (PAT)	3.03	1.60	1.49	0.42	0.36	0.83
human (PAT)	3.36	1.56	1.49	0.37	0.33	0.94

Animal power

Fundamental biological and ecological constraint:

$$\boxed{P = c\,M^{\alpha}}$$

$P =$ basal metabolic rate

$M =$ organismal body mass

Animal power

Fundamental biological and ecological constraint:

$$\boxed{P = c\, M^{\alpha}}$$

$$P = \text{basal metabolic rate}$$

$$M = \text{organismal body mass}$$

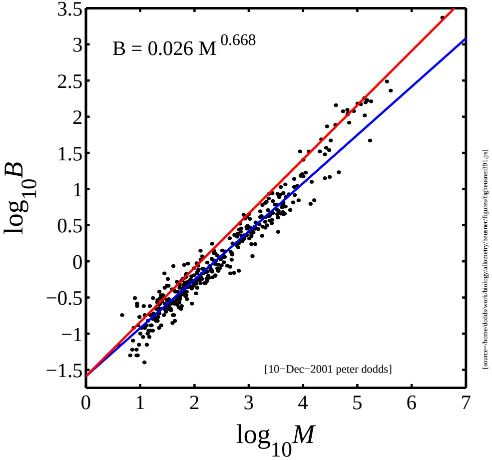
History

1964: Troon, Scotland:
3rd symposium on energy metabolism.
$\alpha = 3/4$ made official . . .

History

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

Some data on metabolic rates

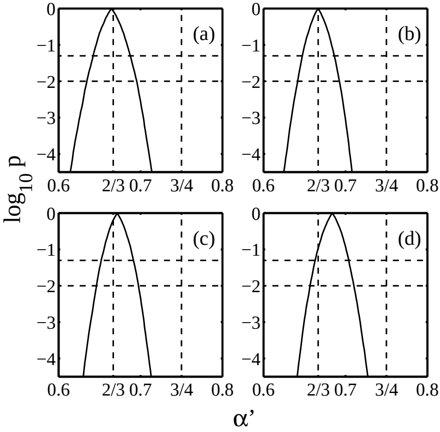


- Heusner's data (1991) [11]
- 391 Mammals
- blue line: 2/3
- red line: 3/4.
- ($B = P$)

Some regressions from the ground up...

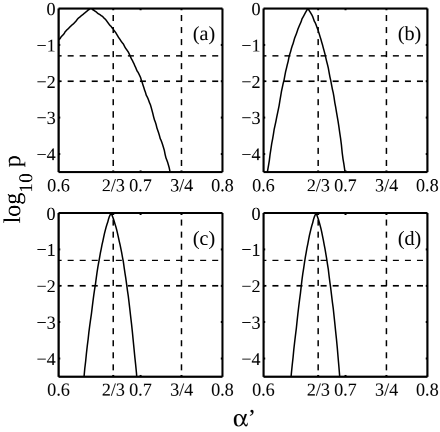
range of M	N	$\hat{\alpha}$
≤ 0.1 kg	167	0.678 ± 0.038
≤ 1 kg	276	0.662 ± 0.032
≤ 10 kg	357	0.668 ± 0.019
≤ 25 kg	366	0.669 ± 0.018
≤ 35 kg	371	0.675 ± 0.018
≤ 350 kg	389	0.706 ± 0.016
≤ 3670 kg	391	0.710 ± 0.021

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

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.
- 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?

Optimal source allocation

Solidifying the basic problem

- Given a region with some population distribution ρ, 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 ρ, 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 ρ, 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

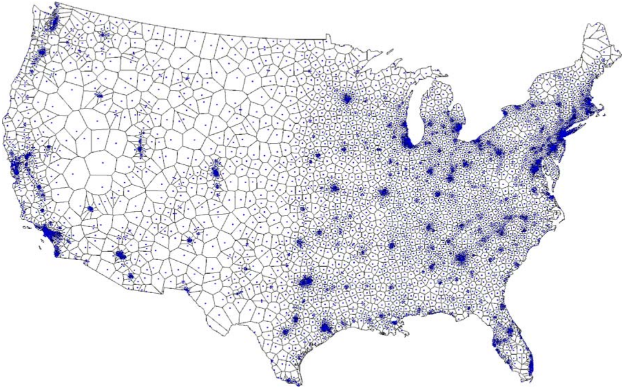
- Given a region with some population distribution ρ, most likely uneven.
- Given resources to build and maintain N facilities.
- Q: How do we locate these N facilities so as to minimize the average distance between an individual's residence and the nearest facility?
- Problem of interested and studied 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 ρ, 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.

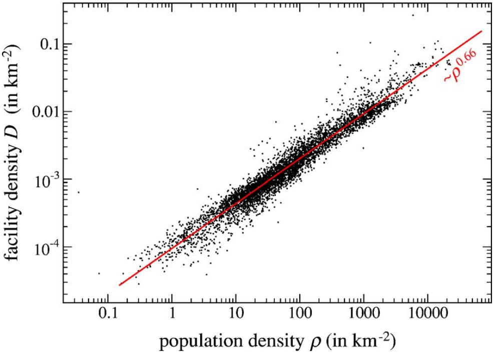
Optimal source allocation



From Gastner and Newman (2006) [8]

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

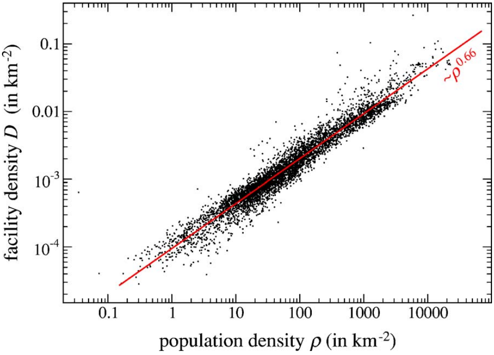
Optimal source allocation



From Gastner and Newman (2006) [8]

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

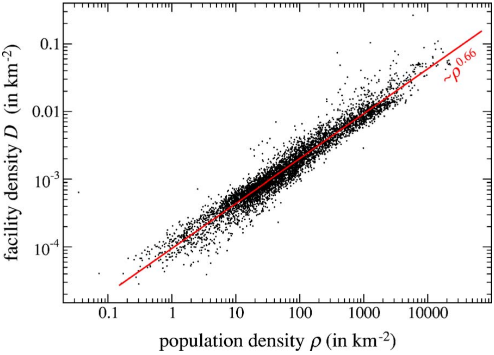
Optimal source allocation



From Gastner and Newman (2006) [8]

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

Outline

Optimal source allocation

Size-density law:

- $$\boxed{D \propto \rho^{2/3}}$$
- In d dimensions:

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

- 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:

- $$\boxed{D \propto \rho^{2/3}}$$
- In d dimensions:

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

- 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:

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

- $$\boxed{D \propto \rho^{2/3}}$$
- In d dimensions:

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

- 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:

- $$\boxed{D \propto \rho^{2/3}}$$
- In d dimensions:

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

- 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

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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

- 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 ρ defined on a spatial region Ω.
- Formally, we want to find the locations of n sources $\{\vec{x}_1, \ldots, \vec{x}_n\}$ that minimizes the cost function

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

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

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

- 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 ith Voronoi cell.
- Sneakiness: set $c_i = c$.
- Compute $\delta G / \delta A$, 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

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

- 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 ith Voronoi cell.
- Sneakiness: set $c_i = c$.
- Compute $\delta G / \delta A$, 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 $\{\vec{x}_1, ..., \vec{x}_n\}$, minimize

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

- 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 ith Voronoi cell.
- Sneakiness: set $c_i = c$.
- Compute $\delta G / \delta A$, 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 $\{\vec{x}_1, ..., \vec{x}_n\}$, minimize

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

- 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 ith Voronoi cell.
- Sneakiness: set $c_i = c$.
- Compute $\delta G / \delta A$, 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

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

- 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 ith Voronoi cell.
- Sneakiness: set $c_i = c$.
- Compute $\delta G/\delta A$, the functional derivative (⊞).
- Solve and substitute $D = 1/A$, we find

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

Outline

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 ℓ_{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 ℓ_{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 ℓ_{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:

$$C_{\mathrm{maint}} + \gamma C_{\mathrm{travel}}.$$

- Travel time is more complicated: Take 'distance' between nodes to be a composite of shortest path distance ℓ_{ij} and number of legs to journey:

$$(1-\delta)\ell_{ij} + \delta(\#\mathrm{hops}).$$

- 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:

$$C_{\mathrm{maint}} + \gamma C_{\mathrm{travel}}.$$

- Travel time is more complicated: Take 'distance' between nodes to be a composite of shortest path distance ℓ_{ij} and number of legs to journey:

$$(1-\delta)\ell_{ij} + \delta(\#\mathrm{hops}).$$

- 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:

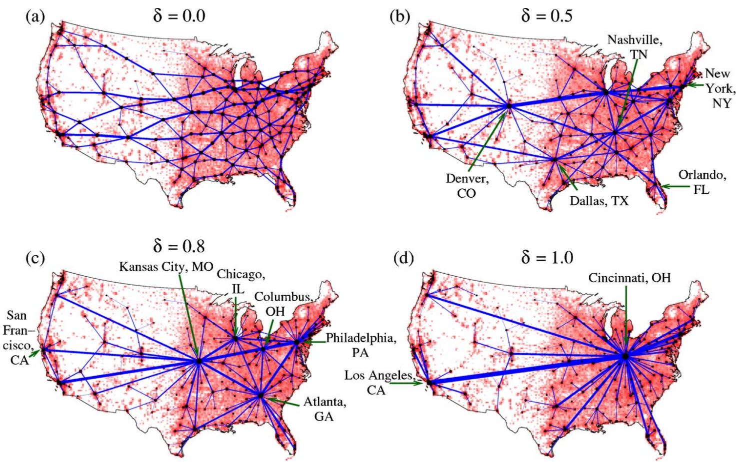
$$C_{\mathrm{maint}} + \gamma C_{\mathrm{travel}}.$$

- Travel time is more complicated: Take 'distance' between nodes to be a composite of shortest path distance ℓ_{ij} and number of legs to journey:

$$(1-\delta)\ell_{ij} + \delta(\#\mathrm{hops}).$$

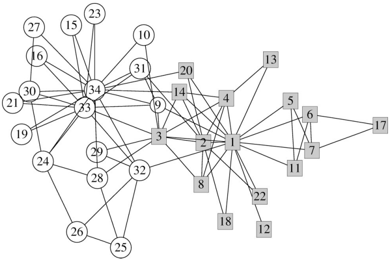
- When $\delta = 1$, only number of hops matters.

Global redistribution networks



From Gastner and Newman (2006) [8]

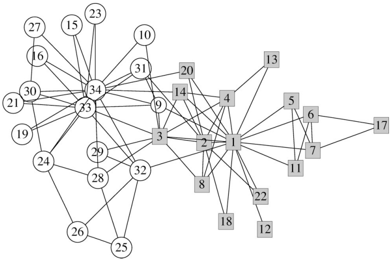
Structure detection



▲ Zachary's karate club [25, 16]

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

Structure detection

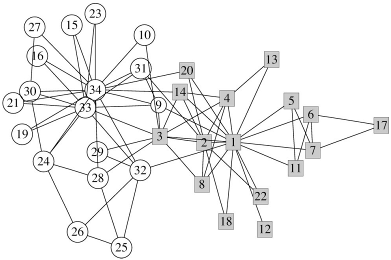


▲ Zachary's karate club [25, 16]

- 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

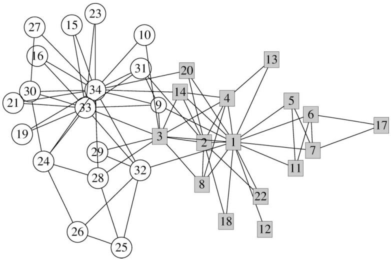


▲ Zachary's karate club [25, 16]

- 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 [25, 16]

- 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...

Outline

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
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[14, 15]
  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.
- Following comes from "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[16]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[14, 15]
  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.
- Following comes from "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[16]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[14, 15]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002).[9]

Hierarchy by division

Top down:

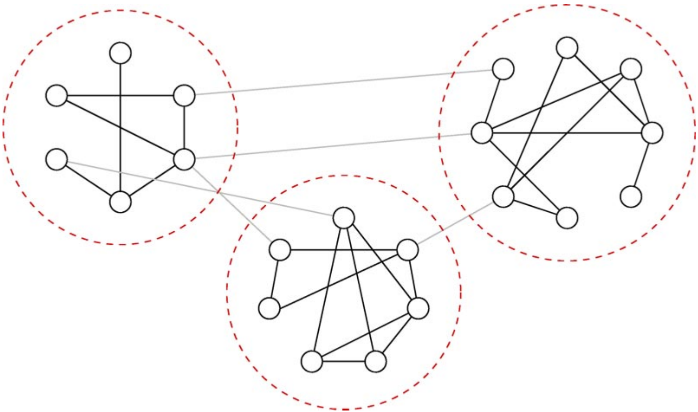
- Idea: Identify global structure first and recursively uncover more detailed structure.
- Basic objective: find dominant components that have significantly more links within than without, as compared to randomized version.
- Following comes from "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[16]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[14, 15]
  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.
- Following comes from "Finding and evaluating community structure in networks" by Newman and Girvan (PRE, 2004).[16]
- See also
  1. "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality" by Newman (PRE, 2001).[14, 15]
  2. "Community structure in social and biological networks" by Girvan and Newman (PNAS, 2002).[9]

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.

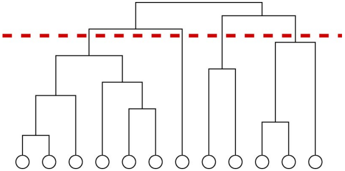
Hierarchy by division

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.

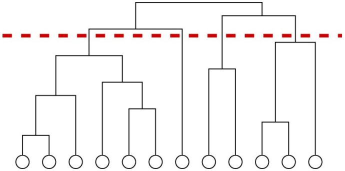
Hierarchy by division

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:

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

  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:

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

where e_{ij} is the fraction of edges between 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:

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

  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:

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

  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:

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

  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:

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

  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:

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

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:

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

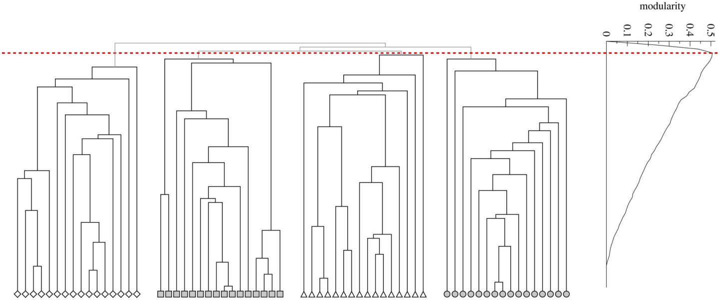
Hierarchy by division

Test case:

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

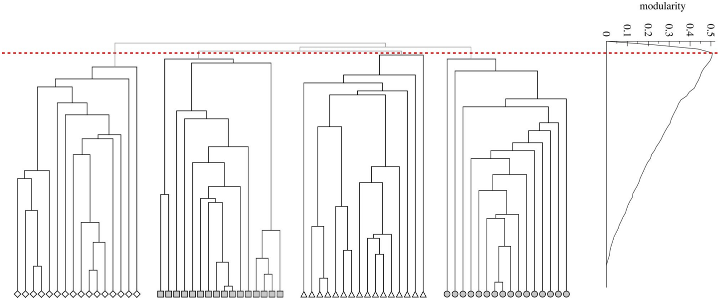
$$\langle k \rangle_{\text{in}} = 6 \text{ and } \langle k \rangle_{\text{out}} = 2.$$

Hierarchy by division



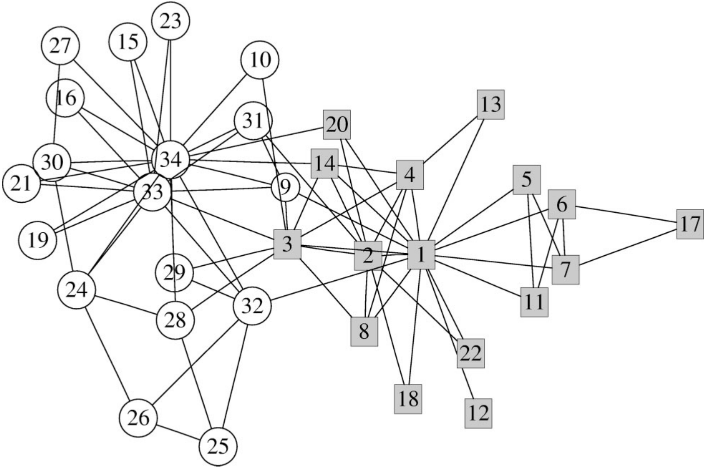
- Maximum modularity $Q \simeq 0.5$ obtained when four communities are uncovered.
- Further 'discovery' of internal structure is somewhat meaningless, as any communities arise accidentally.

Hierarchy by division



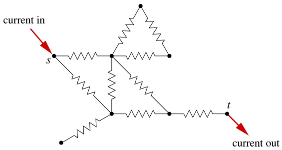
- Maximum modularity $Q \simeq 0.5$ obtained when four communities are uncovered.
- Further ‘discovery’ of internal structure is somewhat meaningless, as any communities arise accidentally.

Hierarchy by division



- Factions in Zachary's karate club network. [25]

Betweenness for electrons:

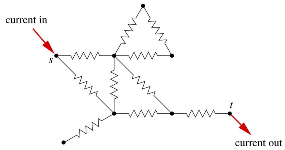


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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

$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

Betweenness for electrons:

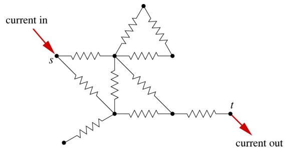


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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

$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

Betweenness for electrons:

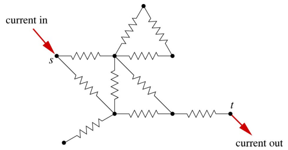


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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

$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

Betweenness for electrons:

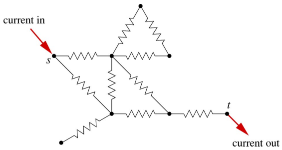


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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

$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

Betweenness for electrons:

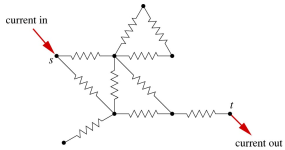


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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



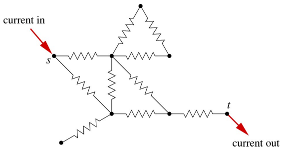
- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.

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

$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

Betweenness for electrons:

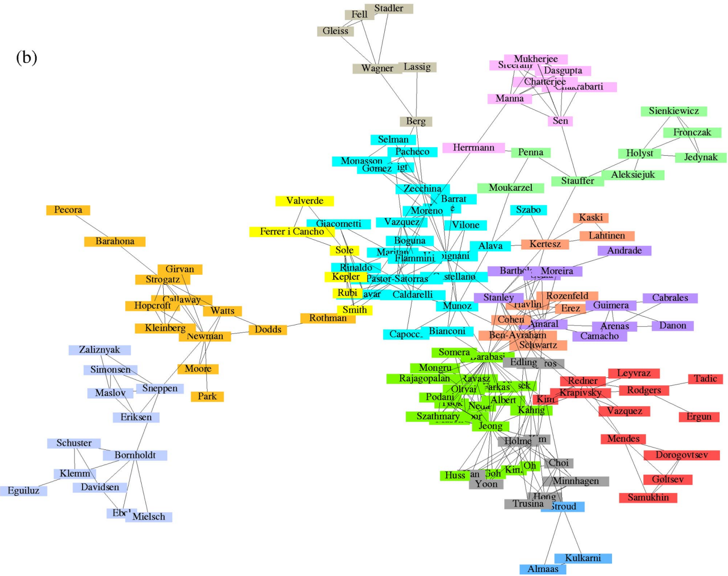


- Unit resistors on each edge.
- For every pair of nodes s (source) and t (sink), set up unit currents in at s and out at t.
- Measure absolute current along each edge ℓ, $|I_{\ell,st}|$.
- Sum $|I_{\ell,st}|$ over all pairs of nodes to obtain electronic betweenness for edge ℓ.
- (Equivalent to random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

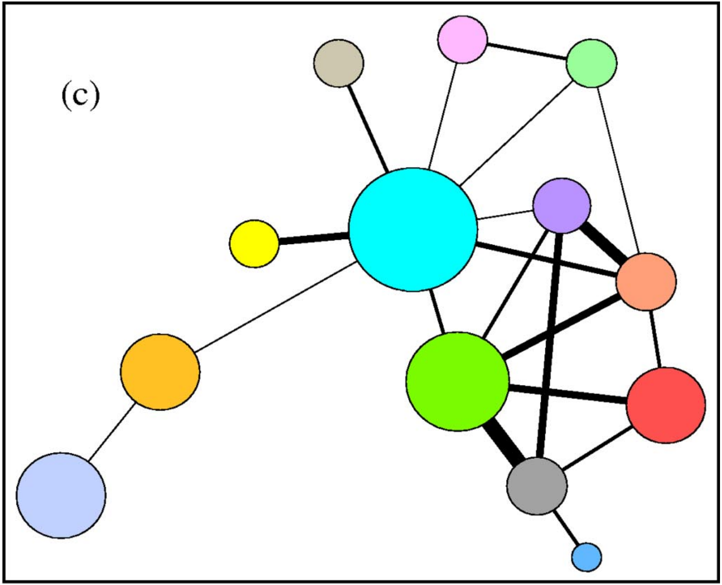
$$B_{ij}^{\mathrm{elec}} = a_{ij}|V_i - V_j|.$$

- Upshot: specific measure of betweenness not too important.

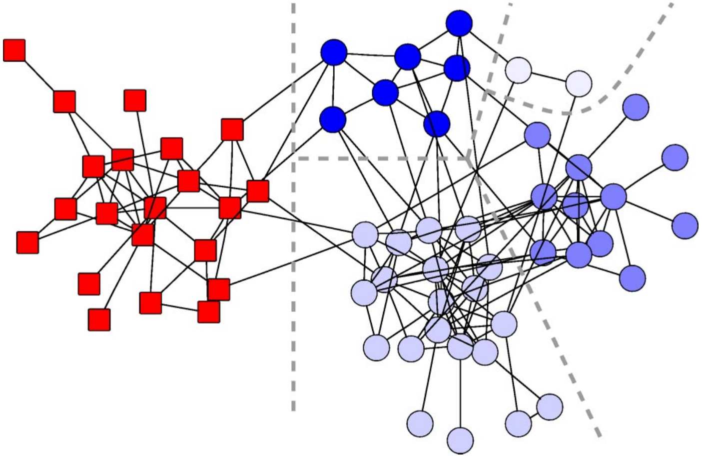
Scientists working on networks



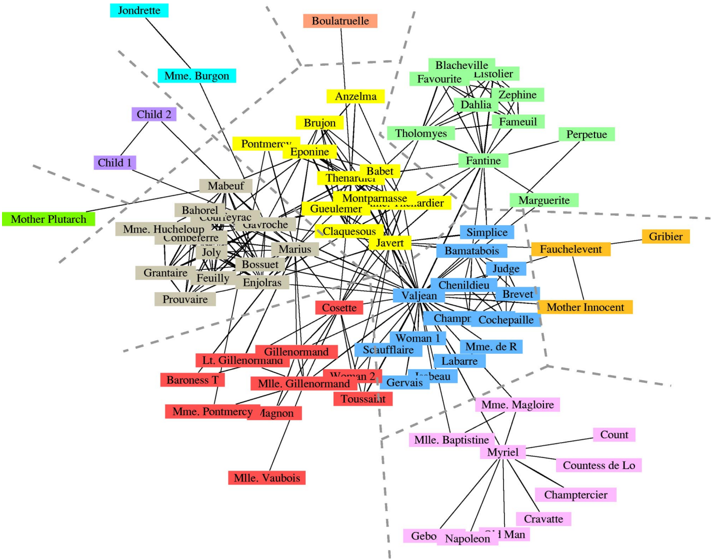
Scientists working on networks



Dolphins!



Les Miserables



Outline

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:

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

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:

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

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:

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

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} = **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** 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.

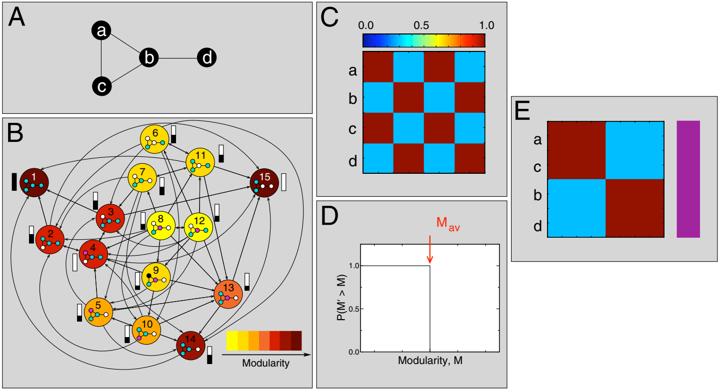
Shuffling for structure

- 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

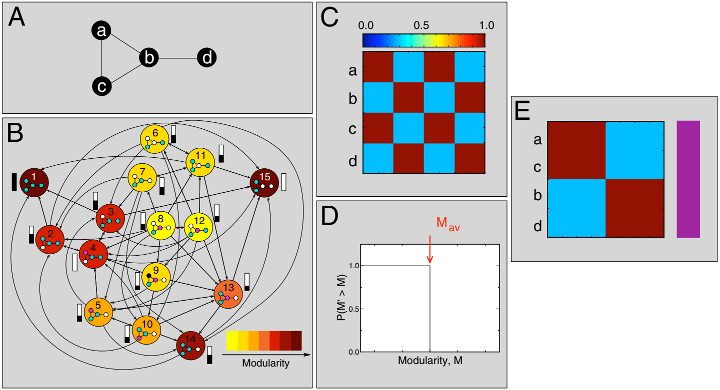
- Consider partition network, i.e., the network of all possible partitions.
- Defn: Two partitions are connected if they differ only by the reassignment of a single node.
- Look for local maxima in partition network.
- Construct an affinity matrix with entries A_{ij}.
- $A_{ij} = \mathbf{Pr}$ random 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...

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).

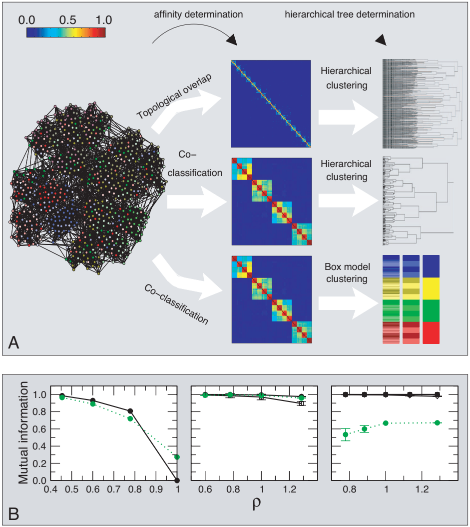
Shuffling for structure

- 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

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

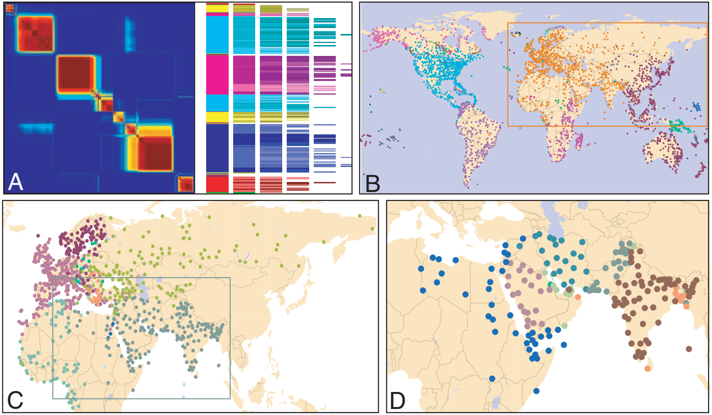
- Use simulated annealing (slow).

Shuffling for structure



- $N = 640$,
- $\langle k \rangle = 16$,
- 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

- "Detecting communities in large networks" Capocci *et al.* (2005) [4]
- Consider normal matrix $\mathbf{K}^{-1}A$, random walk matrix $A^{\mathrm{T}}\mathbf{K}^{-1}$, Laplacian $\mathbf{K} - \mathbf{A}$, 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

- "Detecting communities in large networks"
  Capocci *et al.* (2005) [4]
- Consider normal matrix $\mathbf{K}^{-1}A$, random walk matrix $A^{\mathrm{T}}\mathbf{K}^{-1}$, Laplacian $\mathbf{K} - \mathbf{A}$, 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

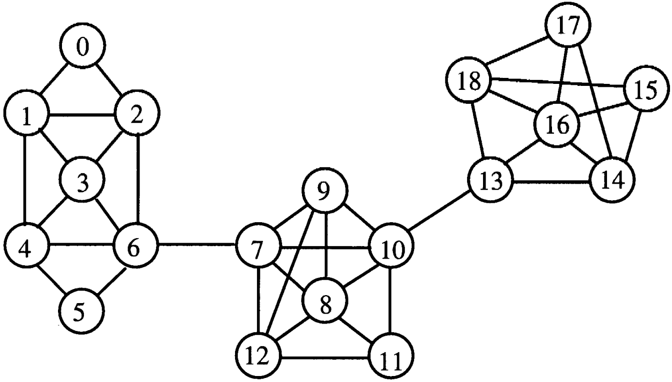
- "Detecting communities in large networks" Capocci *et al.* (2005) [4]
- Consider normal matrix $\mathbf{K}^{-1}A$, random walk matrix $A^{\mathrm{T}}\mathbf{K}^{-1}$, Laplacian $\mathbf{K}-\mathbf{A}$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Build on Kleinberg's HITS algorithm. [13]

General structure detection

- "Detecting communities in large networks" Capocci *et al.* (2005) [4]
- Consider normal matrix $\mathbf{K}^{-1}A$, random walk matrix $A^{\mathrm{T}}\mathbf{K}^{-1}$, Laplacian $\mathbf{K} - \mathbf{A}$, and AA^{T}.
- Basic observation is that eigenvectors associated with secondary eigenvalues reveal evidence of structure.
- Build on Kleinberg's HITS algorithm. [13]

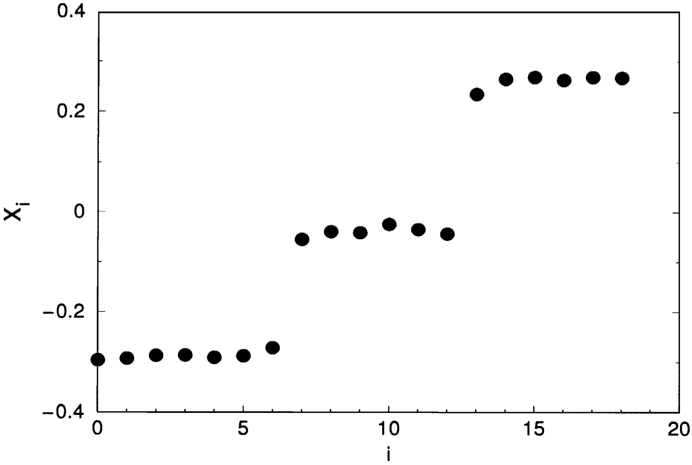
General structure detection

- Example network:



General structure detection

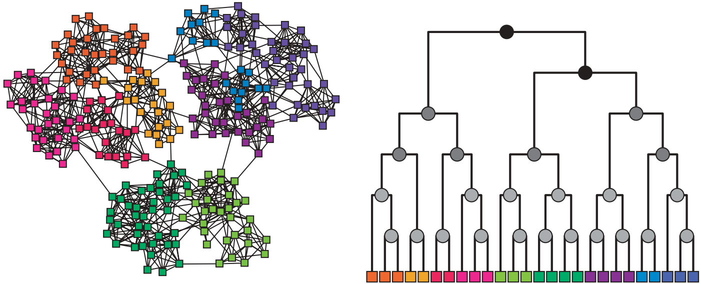
- Second eigenvector's components:



Outline

Hierarchies and missing links

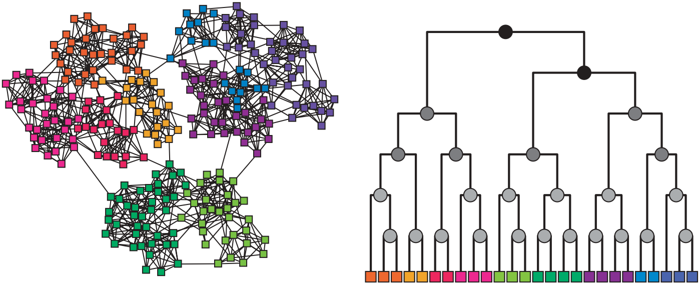
Clauset *et al.*, Nature (2008) [5]



- 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

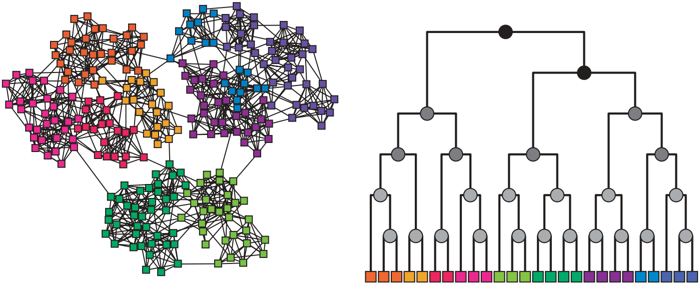
Clauset *et al.*, Nature (2008) [5]



- 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

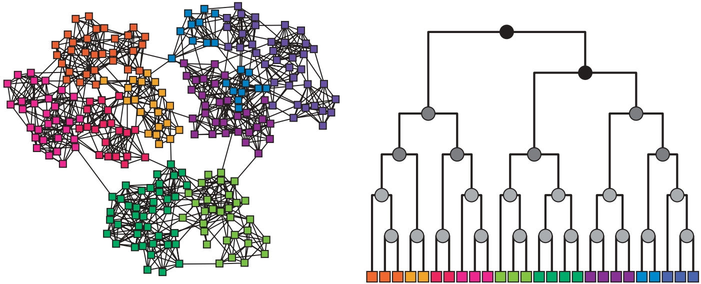
Clauset *et al.*, Nature (2008) [5]



- 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

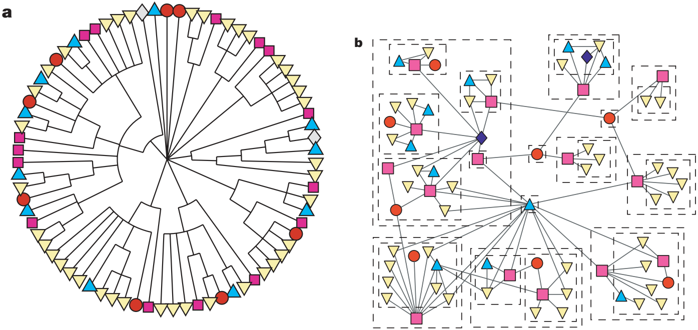
- Model also predicts reasonably well
  1. average degree,
  2. clustering,
  3. and average shortest path length.

Table 1 | Comparison of original and resampled networks

Network	$\langle k \rangle_{real}$	$\langle k \rangle_{samp}$	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)

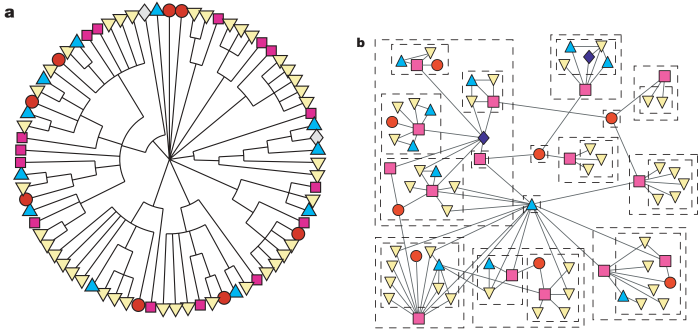
Statistics are shown for the three example networks studied and for new networks generated by resampling from our hierarchical model. The generated networks closely match the average degree $\langle k \rangle$, clustering coefficient C and average vertex–vertex distance d in each case, suggesting that 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

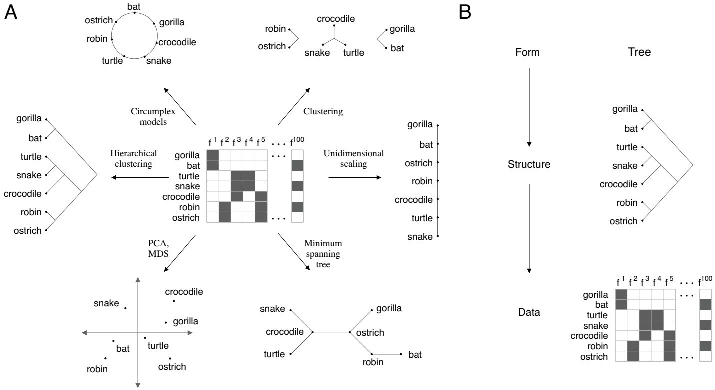


- Consensus dendogram for grassland species.
- Copes with disassortative and assortative communities.

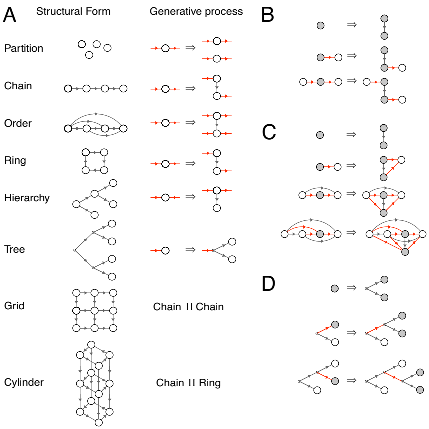
Outline

General structure detection

- "The discovery of structural form"
  Kemp and Tenenbaum, PNAS (2008) [12]

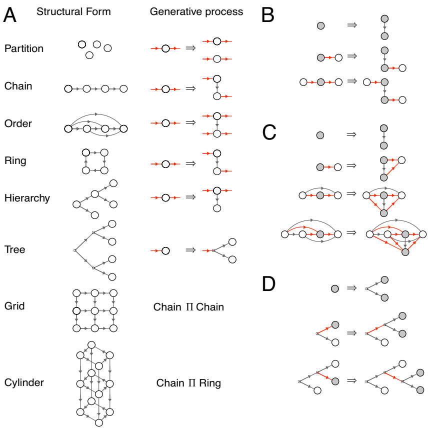


General structure detection



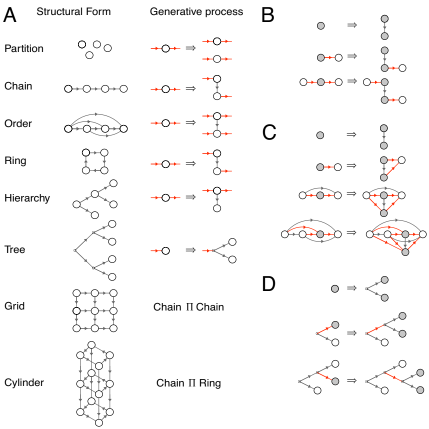
- Top down description of form.
- Node replacement graph grammar: parent node becomes two child nodes.
- B-D: Growing chains, orders, and trees.

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.

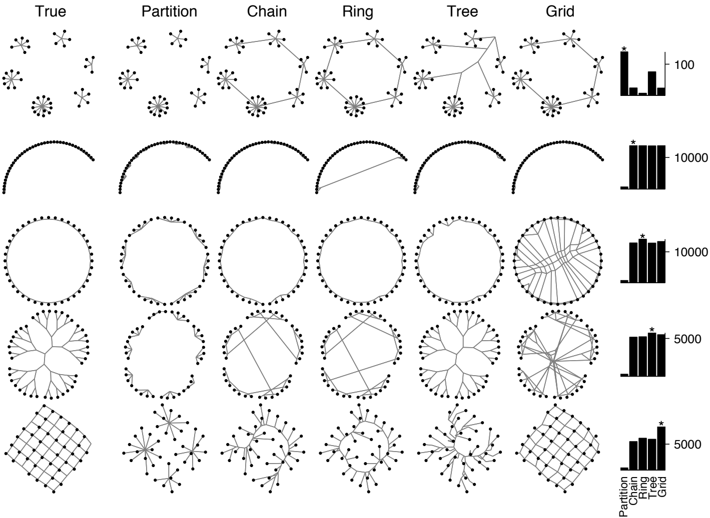
General structure detection



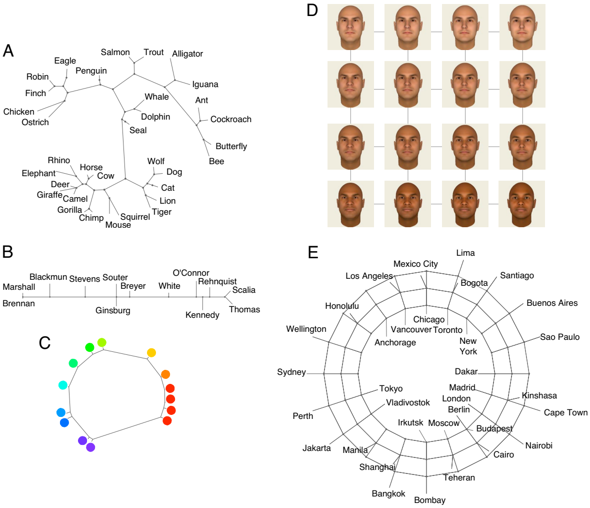
- Top down description of form.
- Node replacement graph grammar: parent node becomes two child nodes.
- B-D: Growing chains, orders, and trees.

General structure detection

- Performance for test networks.

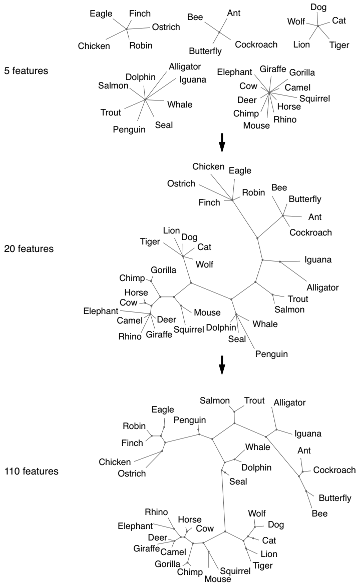


Example learned structures:



- 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

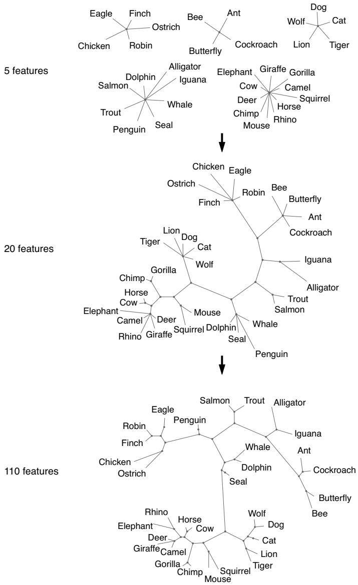
$\Downarrow$

simple tree

$\Downarrow$

complex tree

General structure detection



- Effect of adding features on detected form.

Straight partition

$\Downarrow$

simple tree

$\Downarrow$

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

[1] J. R. Banavar, F. Colaiori, A. Flammini, A. Maritan, and A. Rinaldo.
Topology of the fittest transportation network.
Phys. Rev. Lett., 84:4745–4748, 2000. pdf (⊞)

[2] J. R. Banavar, A. Maritan, and A. Rinaldo.
Size and form in efficient transportation networks.
Nature, 399:130–132, 1999. pdf (⊞)

[3] S. Bohn and M. O. Magnasco.
Structure, scaling, and phase transition in the optimal transport network.
Phys. Rev. Lett., 98:088702, 2007. pdf (⊞)

References II

[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 (⊞)

[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] P. S. Dodds and D. H. Rothman.
Unified view of scaling laws for river networks.
Physical Review E, 59(5):4865–4877, 1999. pdf (⊞)

References III

[7] P. S. Dodds and D. H. Rothman.
Geometry of river networks. I. Scaling, fluctuations, and deviations.
Physical Review E, 63(1):016115, 2001. pdf (⊞)

[8] M. T. Gastner and M. E. J. Newman.
Optimal design of spatial distribution networks.
Phys. Rev. E, 74:016117, 2006. pdf (⊞)

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

[10] S. M. Gusein-Zade.
Bunge's problem in central place theory and its generalizations.
Geogr. Anal., 14:246–252, 1982.

References IV

[11] A. A. Heusner.
Size and power in mammals.
Journal of Experimental Biology, 160:25–54, 1991.

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

[13] J. M. Kleinberg.
Authoritative sources in a hyperlinked environment.
Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. pdf (⊞)

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

References V

[15] 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 (⊞)

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

[17] 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 (⊞)

References VI

[18] 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 (⊞)

[19] G. E. Stephan.
Territorial division: The least-time constraint behind the formation of subnational boundaries.
Science, 196:523–524, 1977. pdf (⊞)

[20] G. E. Stephan.
Territorial subdivision.
Social Forces, 63:145–159, 1984. pdf (⊞)

[21] D. L. Turcotte, J. D. Pelletier, and W. I. Newman.
Networks with side branching in biology.
Journal of Theoretical Biology, 193:577–592, 1998.

References VII

[22] G. B. West, J. H. Brown, and B. J. Enquist.
A general model for the origin of allometric scaling laws in biology.
Science, 276:122–126, 1997. pdf (⊞)

[23] Q. Xia.
The formation of a tree leaf.
Submitted. pdf (⊞)

[24] Q. Xia.
Optimal paths related to transport problems.
Communications in Contemporary Mathematics, 5:251–279, 2003. pdf (⊞)

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