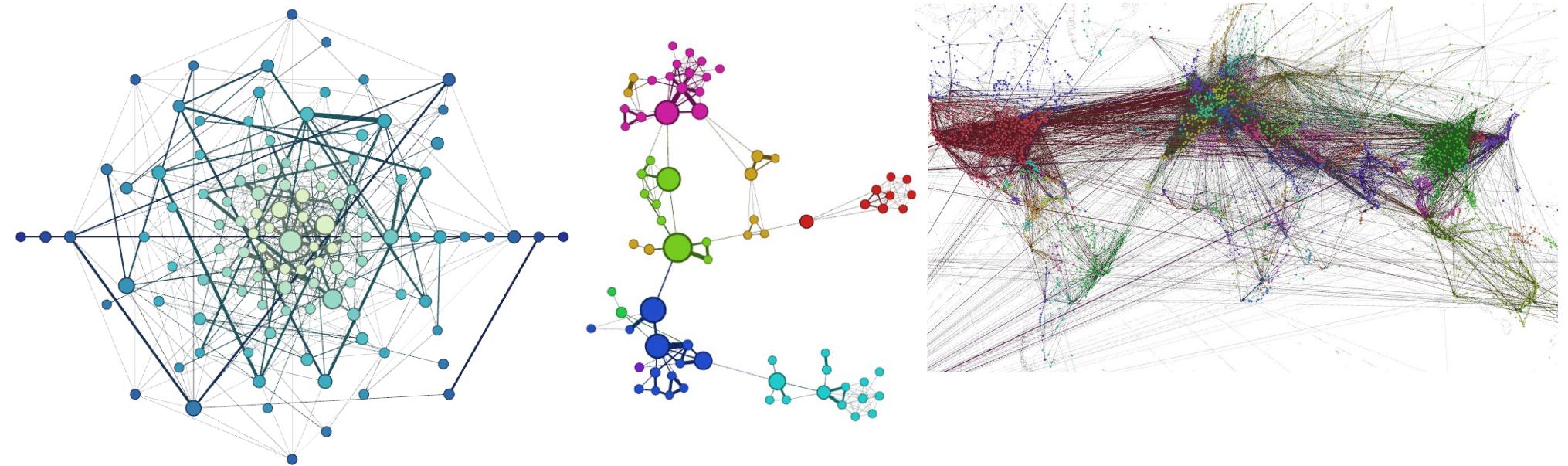


An Introduction To Network Science

U.PORTO
FC FACULDADE DE CIÉNCIAS
UNIVERSIDADE DO PORTO

Pedro Ribeiro
(DCC/FCUP & CRACS/INESC-TEC)



Motivation and the “small world” phenomenon

Planet Earth



8 Billion Humans



How many “degrees” of separation?

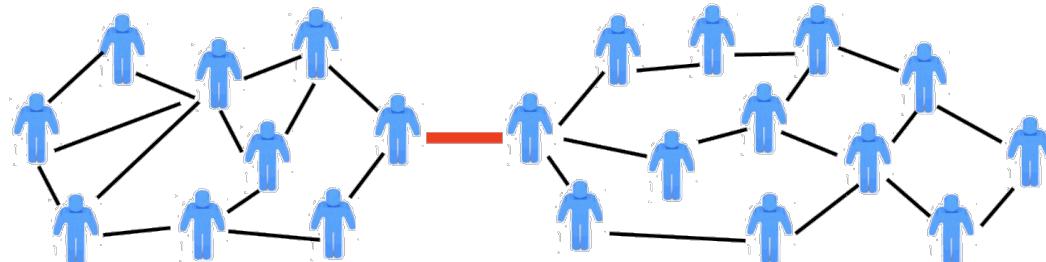




1929

Frigyes Karinthy

“If you choose a person out of the 1.5 billions of our planet, I bet that using no more than **five** individuals, one of them my acquaintance, I could contact the person you chose, using only the list of acquaintances of each one”





1969

Stanley Milgram

- People chosen at random on a US State
- Request to send a letter to a given final person in another state :
 - If you know the final person, send directly to him
 - If not, send to someone you think it is more likely to know him

An Experimental Study of the Small World Problem*

JEFFREY TRAVERS

Harvard University

AND

STANLEY MILGRAM

The City University of New York

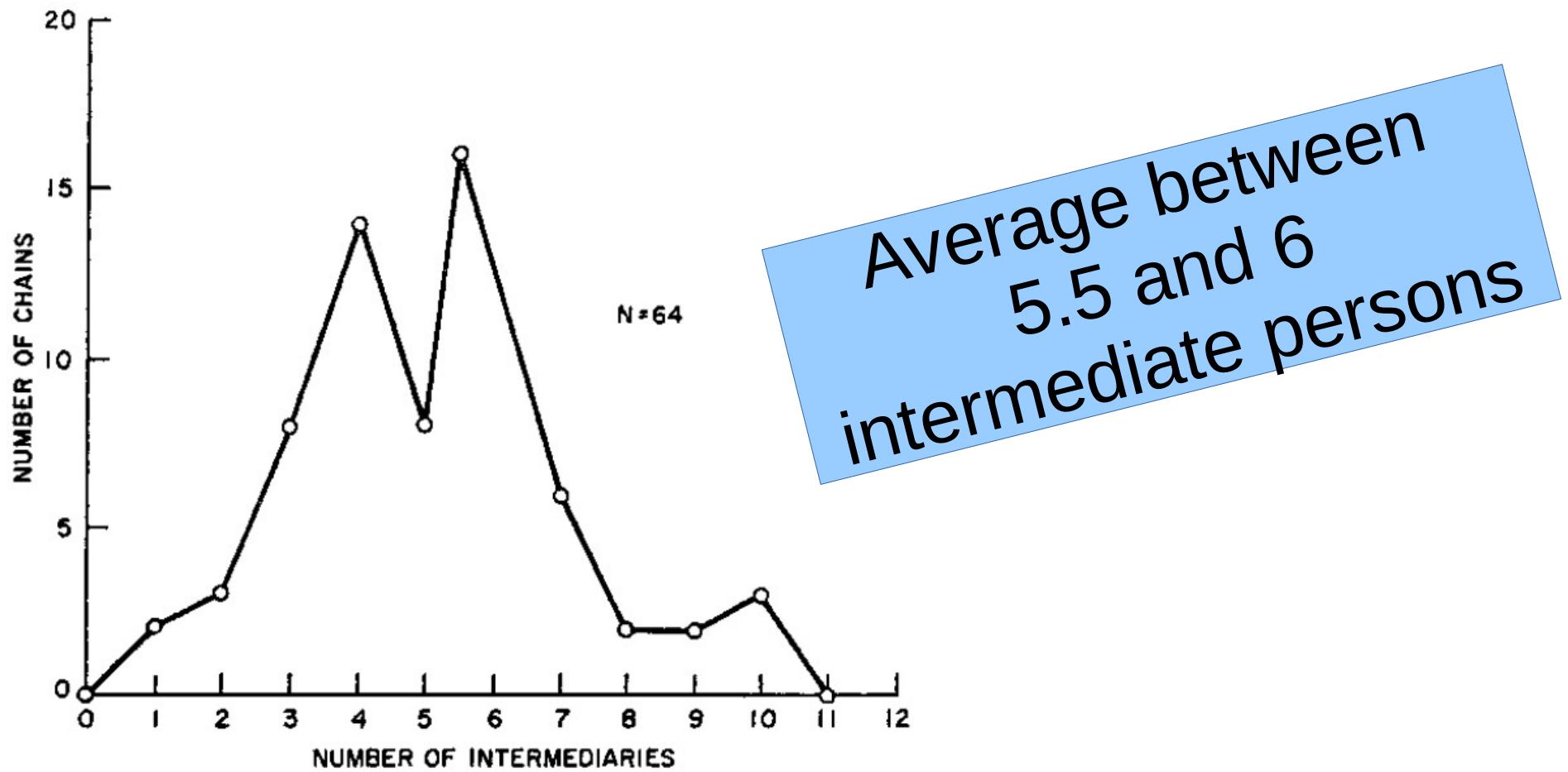
Arbitrarily selected individuals ($N=296$) in Nebraska and Boston are asked to generate acquaintance chains to a target person in Massachusetts, employing "the small world method" (Milgram, 1967). Sixty-four chains reach the target person. Within this group the mean number of intermediaries between starters and targets is 5.2. Boston starting chains reach the target person with fewer intermediaries than those starting in Nebraska; subpopulations in the Nebraska group do not differ among themselves. The funneling of chains through sociometric "stars" is noted, with 48 per cent of the chains passing through three persons before reaching the target. Applications of the method to studies of large scale social structure are discussed.





1969

Stanley Milgram

*Lengths of Completed Chains*

- More than 20.000 chains of emails to 18 persons of 13 countries

An Experimental Study of Search in Global Social Networks

Peter Sheridan Dodds,¹ Roby Muhamad,² Duncan J. Watts^{1,2*}

We report on a global social-search experiment in which more than 60,000 e-mail users attempted to reach one of 18 target persons in 13 countries by forwarding messages to acquaintances. We find that successful social search is conducted primarily through intermediate to weak strength ties, does not require highly connected “hubs” to succeed, and, in contrast to unsuccessful social search, disproportionately relies on professional relationships. By accounting for the attrition of message chains, we estimate that social searches can reach their targets in a median of five to seven steps, depending on the separation of source and target, although small variations in chain lengths and participation rates generate large differences in target reachability. We conclude that although global social networks are, in principle, searchable, actual success depends sensitively on individual incentives.

Median estimated between 5 and 7

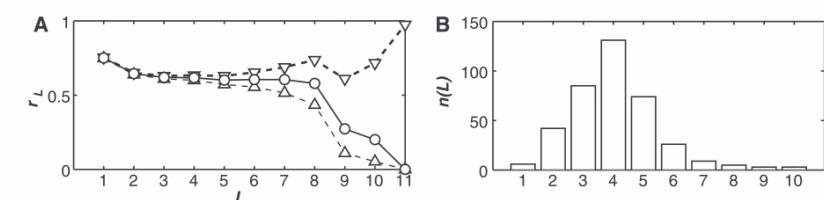
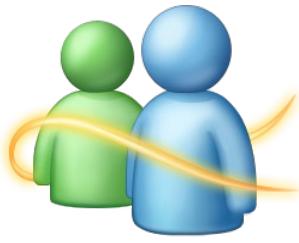


Fig. 1. Distributions of message chain lengths. (A) Average per-step attrition rates (circles) and 95% confidence interval (triangles). (B) Histogram representing the number of chains that are completed in L steps ($\langle L \rangle = 4.01$). (C) “Ideal” histogram of chain lengths recovered from (B) by accounting for message attrition (A). Bars represent the ideal histogram recovered with average values of r [circles in (A)] for the histogram in (B); lines represent a decomposition of the complete data into chains that start in the same country as the target (circles) and those that start in a different country (triangles).



2008

Microsoft Messenger

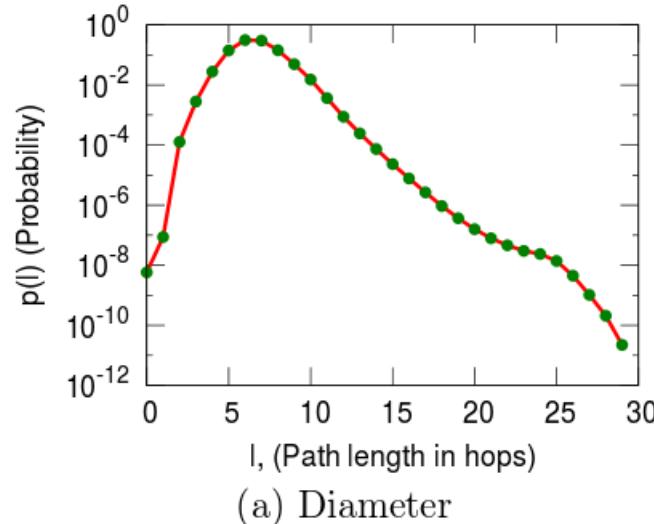
- 30 billion conversations between 240 million persons

Planetary-Scale Views on a Large Instant-Messaging Network

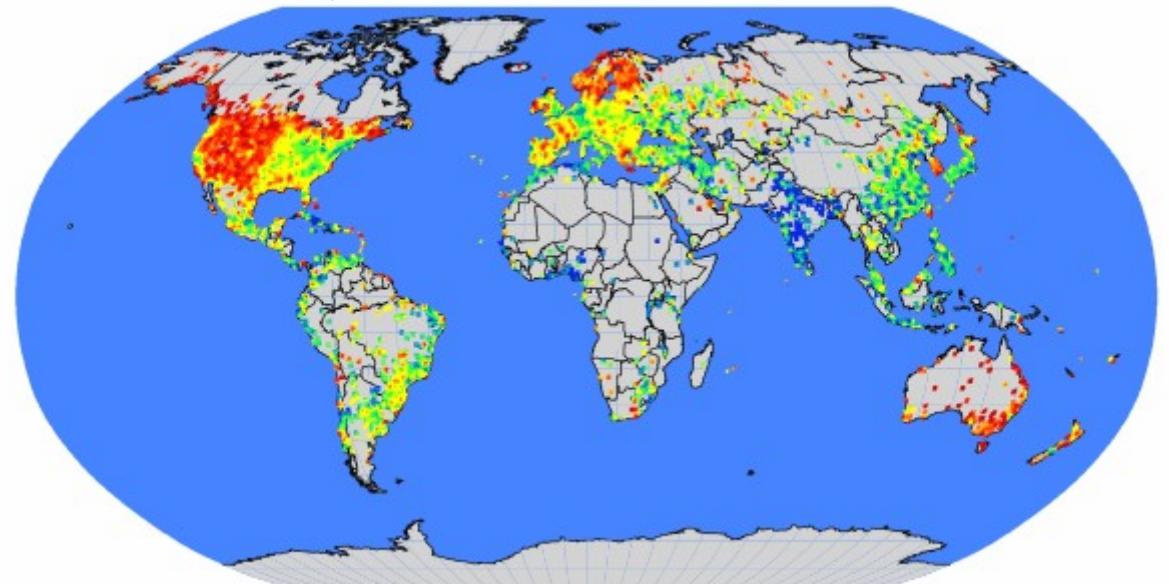
Jure Leskovec *
Carnegie Mellon University
jure@cs.cmu.edu

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

Global Average: 5.6



(a) Diameter

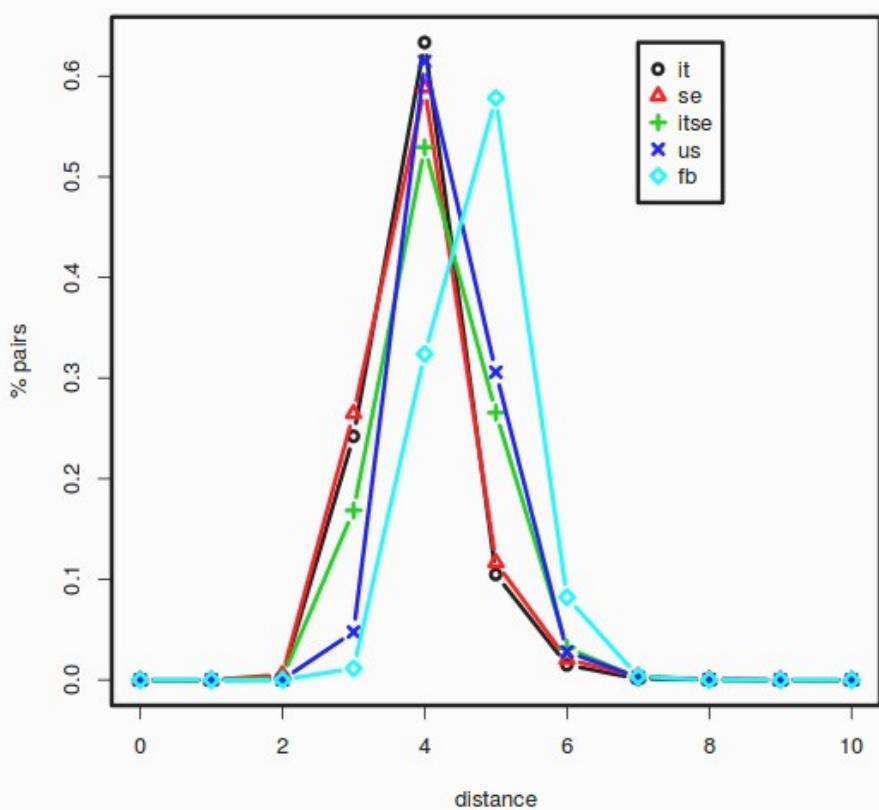




2011

Facebook Friendships

- 69 billions of friendships between 721 millions of persons



Global average: 3.74

Computer Science > Social and Information Networks

Four Degrees of Separation

Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, Sebastiano Vigna

(Submitted on 19 Nov 2011 (v1), last revised 5 Jan 2012 (this version, v3))

Frigyes Karinthy, in his 1929 short story "L'aanczemek" ("Chains") suggested that any two persons are distant individuals, one of whom is a personal acquaintance, he could contact the selected individual [...]. It is not complete graph theory, but the "six degrees of separation" phrase stuck after John Guare's 1990 eponymous play. Follow me, where "distance" is the usual path length-the number of arcs in the path.) Stanley Milgram in his famous experiment average number of intermediaries on the path of the postcards lay between 4.4 and 5.7, depending on the sample. We report the results of the first world-scale social-network graph-distance computations, using the entire Facebook network corresponding to 3.74 intermediaries or "degrees of separation", showing that the world is even smaller than we thought. We explore interesting geographic subgraphs, looking also at their evolution over time. The networks we are able to explore are almost two orders of magnitude larger than those analysed in the previous work, very accurate.



2016

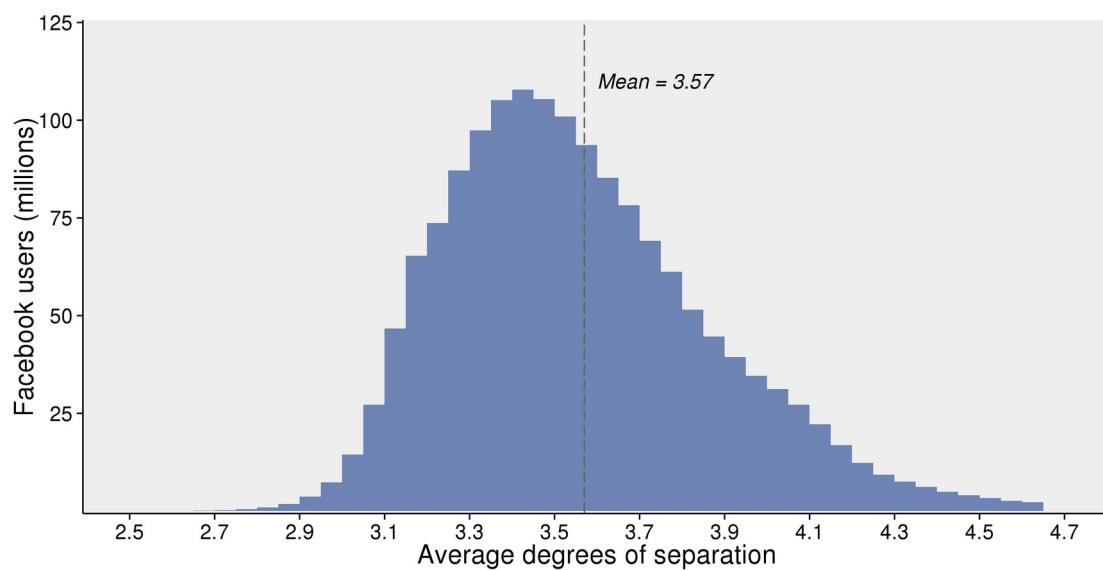
Facebook Friendships

- 1.59 billions of persons

My degrees of separation



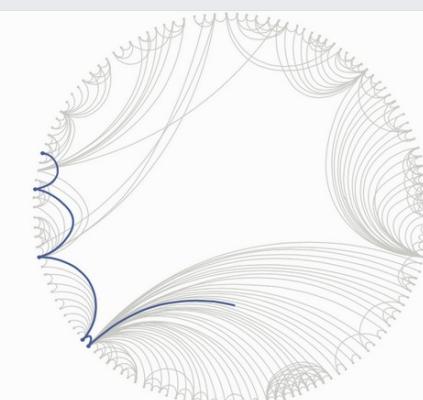
Pedro Ribeiro's average degrees of separation from everyone is 3.43.



Global average: 3.57

Three and a half degrees of separation

Blog



Sergey Edunov, Carlos Diuk, Ismail Onur Filiz, Smriti Bhagat, Moira Burke

4/2
Core Data Science, Social Computing

Gosto Partilhar 63.863

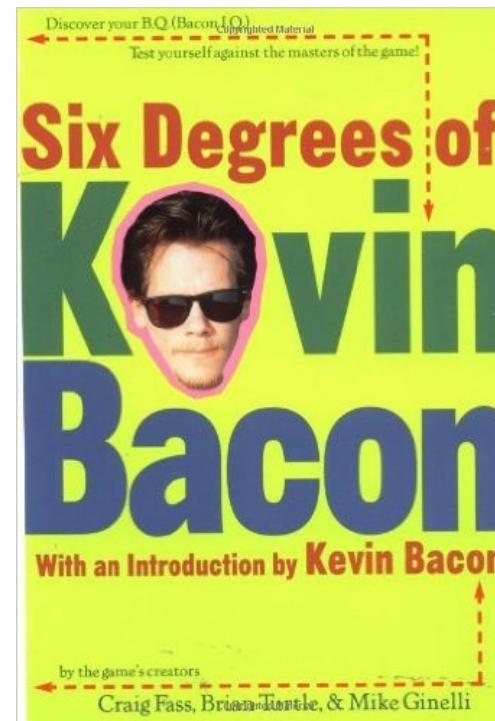
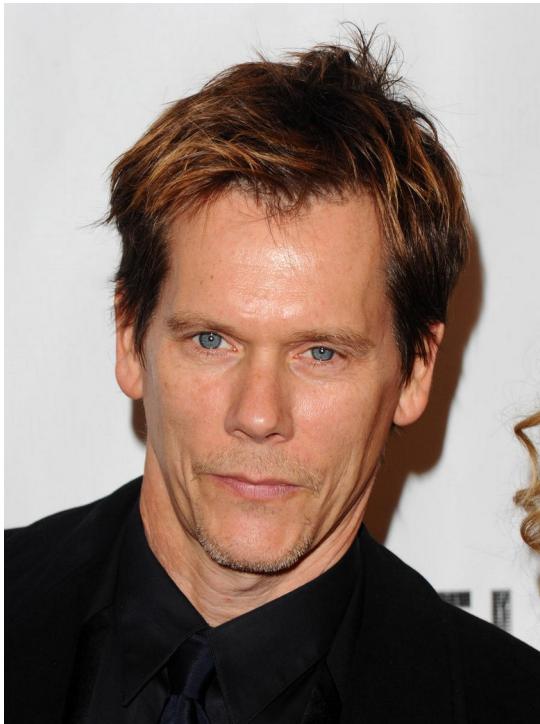
How to explain this?

- Imagine that a person has, on average, 100 friends
 - 0 intermediates: 100
 - 1 intermediate: $100^2 = 10.000$
 - 2 intermediates: $100^3 = 1.000.000$
 - 3 intermediates: $100^4 = 100.000.000$
 - 4 intermediates: $100^5 = 10.000.000.000$
 - 5 intermediates: $100^6 = 1.000.000.000.000$
- In practice, not all friends are new, but still there is a very fast growth

*The power of
exponentiation*

More examples of “Small World”

- The six degrees of Kevin Bacon
 - How many connections to link Kevin Bacon to any other actor, director, producer...
 - “Game” initiated in 1994





*Joaquim
de Almeida*



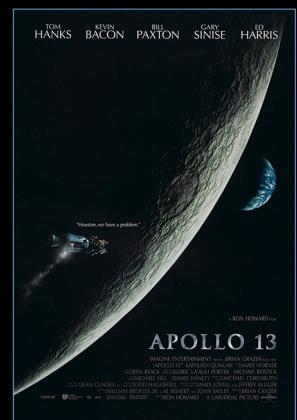
*Joaquim
de Almeida*
(Bacon Number: 2)



Bret Cullen



Kevin Bacon





Joaquim de Almeida

(Bacon Number: 2)

Joaquim de Almeida

was in

The Burning Plain

with

Brett Cullen

was in

Apollo 13

with

Kevin Bacon

Joaquim de Almeida

was in

One Man's Hero

with

Stephen Tobolowsky

was in

Murder in the First

with

Kevin Bacon

Joaquim de Almeida

was in

The Death and Life of Bobby Z

with

Laurence Fishburne

was in

Quicksilver

with

Kevin Bacon

Joaquim de Almeida

was in

Moscow Zero

with

Rade Šerbedžija

was in

X-Men: First Class

with

Kevin Bacon



*Nicolau
Breyner*



Nicolau Breyner

(Bacon Number: 3)

Nicolau Breyner

was in

Night Train to Lisbon

with

Christopher Lee

was in

Alice in Wonderland

with

Michael Sheen

was in

Frost/Nixon

with

Kevin Bacon



*Marilyn
Monroe*



*Charlie
Chaplin*



Marilyn Monroe
(Bacon Number: 2)



Charlie Chaplin
(Bacon Number: 2)



More examples of “Small World”

- The six degrees of Kevin Bacon

<https://oracleofbacon.org/>

(average number: 3.009)

Kevin Bacon Number	# of persons
0	1
1	3150
2	373876
3	1340703
4	340756
5	28820
6	3383
7	451
8	52
9	8
10	1

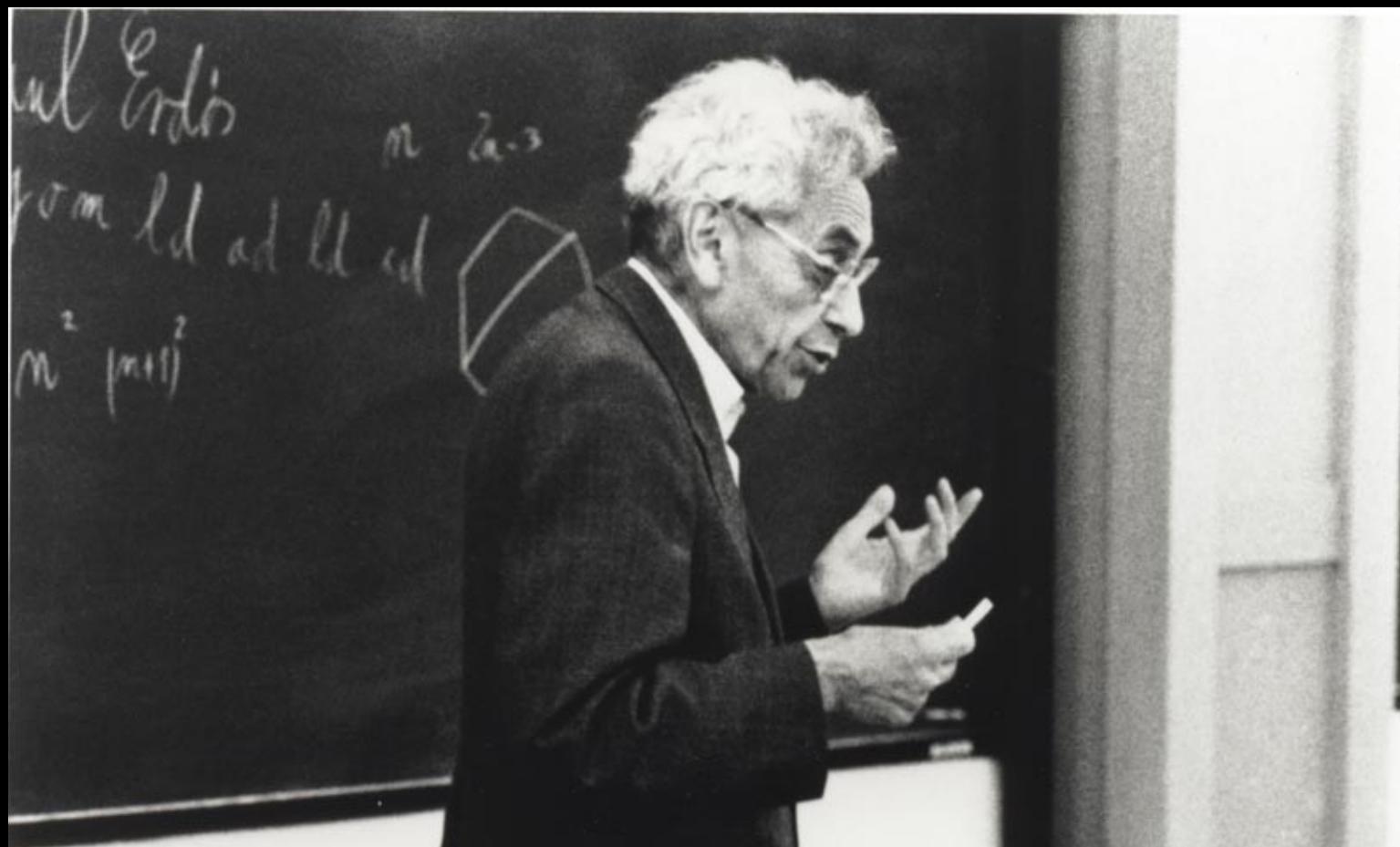


“People would start to come up to me in the subway and literally go...”



“Zero! Zero! Zero! Zero!”

More examples of “Small World”



Paul Erdős

Erdős Number

More examples of “Small World”

- Erdös number:
 - Scientific articles and a very prolific mathematician

<http://wwwp.oakland.edu/enp/>



[Home](#) | [Preferences](#) | [Free Tools](#) |

[Search MSC](#)

[Collaboration Distance](#)

[Current Journals](#)

[Current Publications](#)

MR Erdos Number = 4

Pedro Ribeiro	coauthored with	Srinivasan Parthasarathy ¹	MR3385657
Srinivasan Parthasarathy ¹	coauthored with	Yusu Wang	MR3685725
Yusu Wang	coauthored with	Boris Aronov	MR2347131
Boris Aronov	coauthored with	Paul Erdős ¹	MR1289067

[Change First Author](#)

[Change Second Author](#)

[New Search](#)

Emergence of Network Science

Complexity

A portrait of Stephen Hawking, an English theoretical physicist, cosmologist, and author. He is shown from the chest up, wearing glasses and a dark shirt. He has thinning hair and is looking slightly to the right of the camera.

“I think the
next century
will be the
century of
complexity”

Stephen Hawking (Jan, 2000)

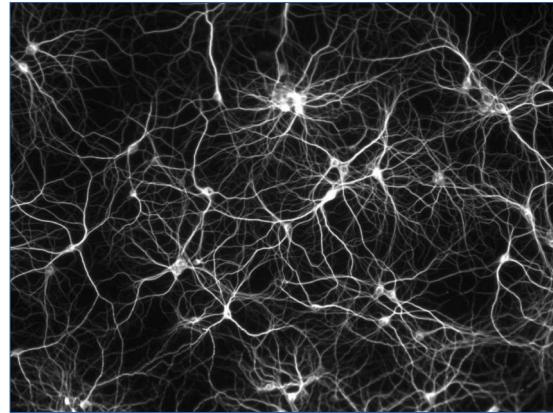
The Real World is Complex

World Population: 8 billions



The Real World is Complex

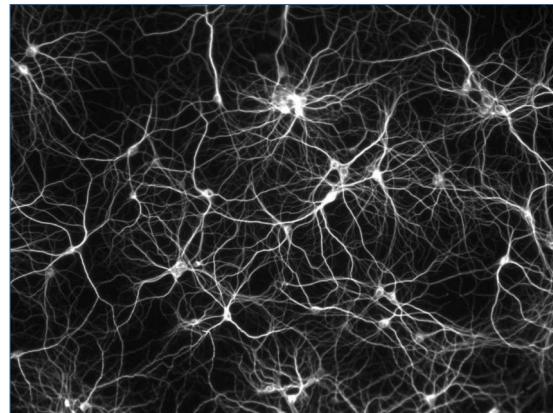
World Population: 8 billions



**Human Brain Neurons:
100 billions**

The Real World is Complex

World Population: 8 billions



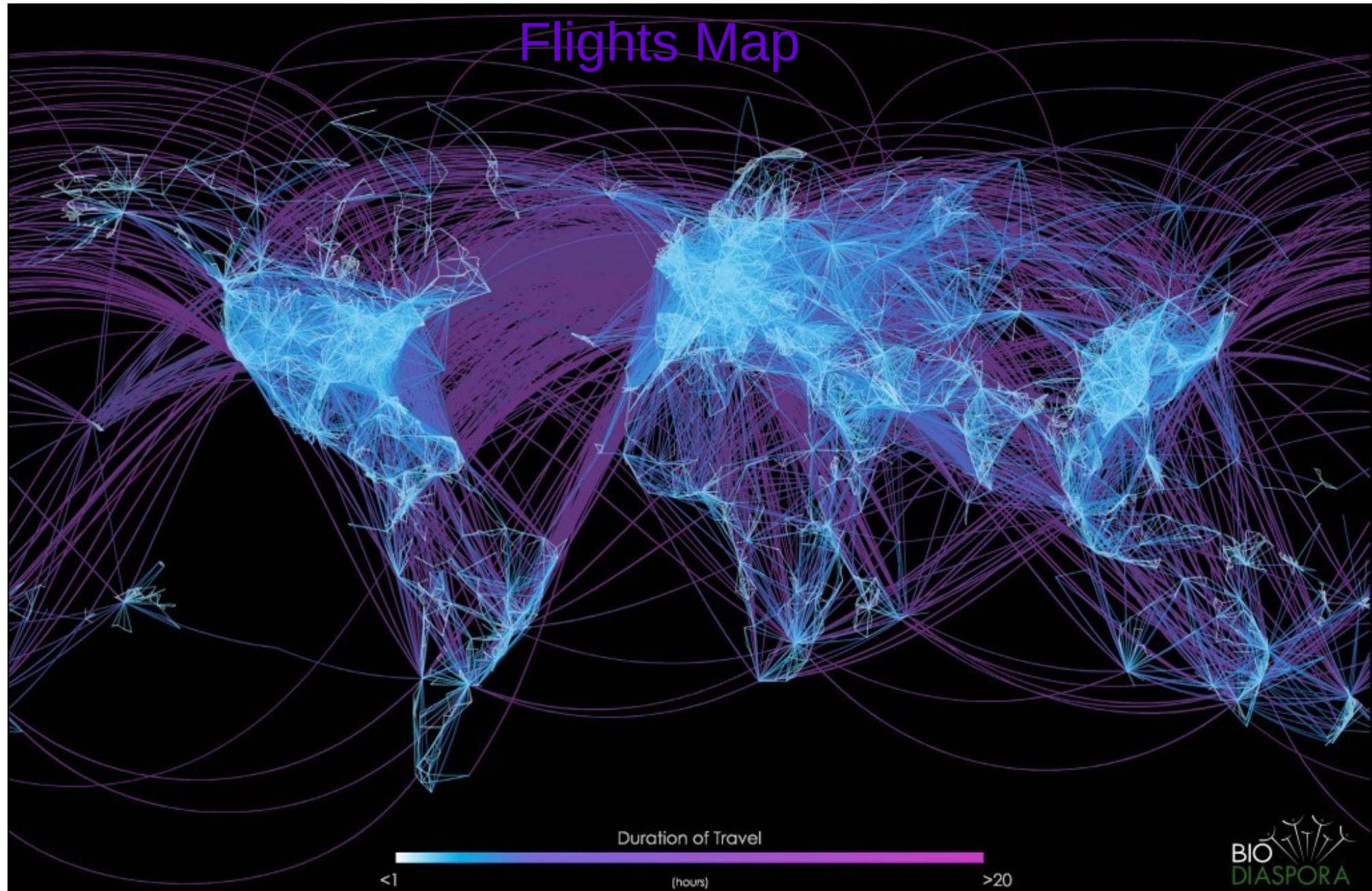
**Human Brain Neurons:
100 billions**

Internet Devices: >8 billions



Complex Systems → Complex Networks

Flights Map

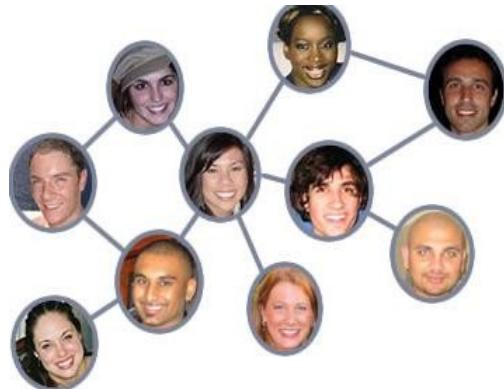


Complex Networks are Ubiquitous

Social

Complex Networks are Ubiquitous

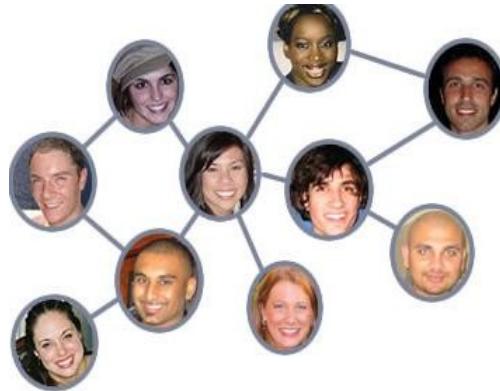
Social



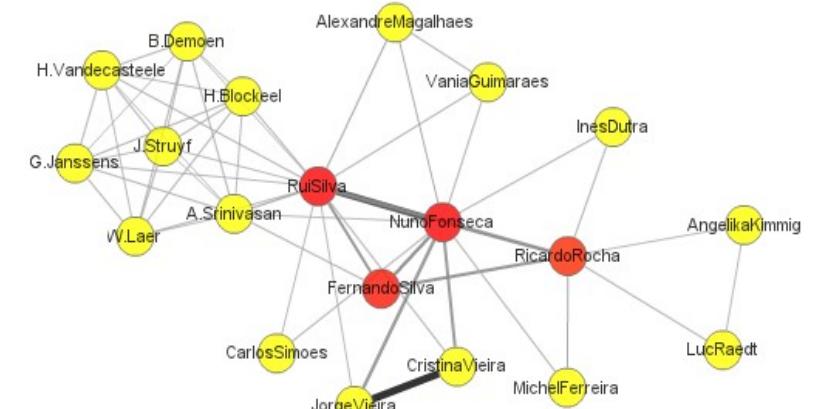
Facebook

Complex Networks are Ubiquitous

Social



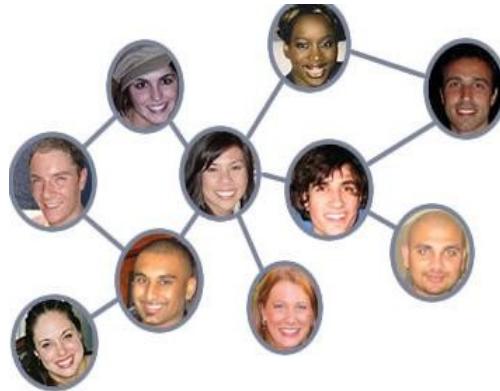
Facebook



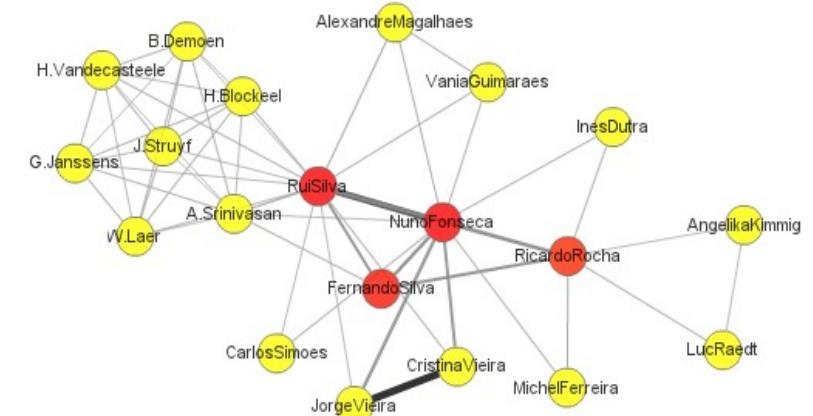
Co-authorship

Complex Networks are Ubiquitous

Social



Facebook

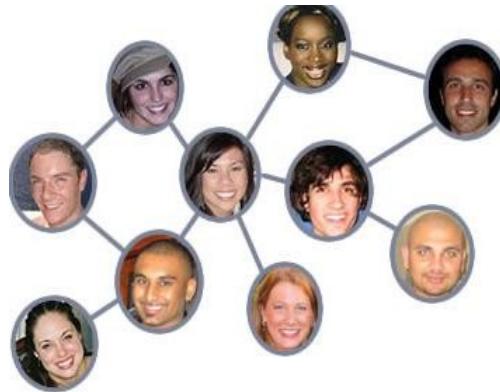


Co-authorship

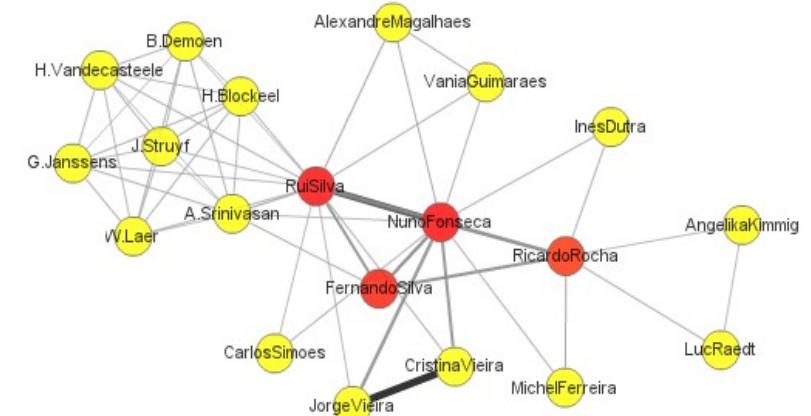
Biological

Complex Networks are Ubiquitous

Social

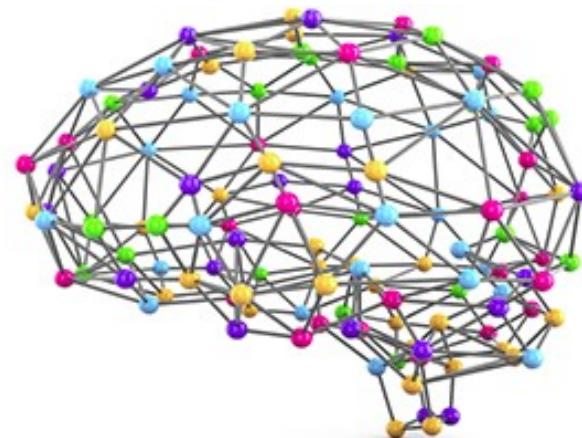


Facebook



Co-authorship

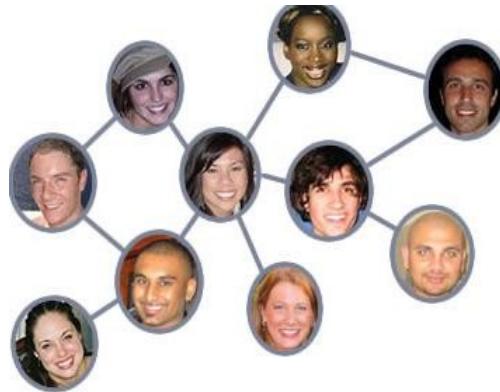
Biological



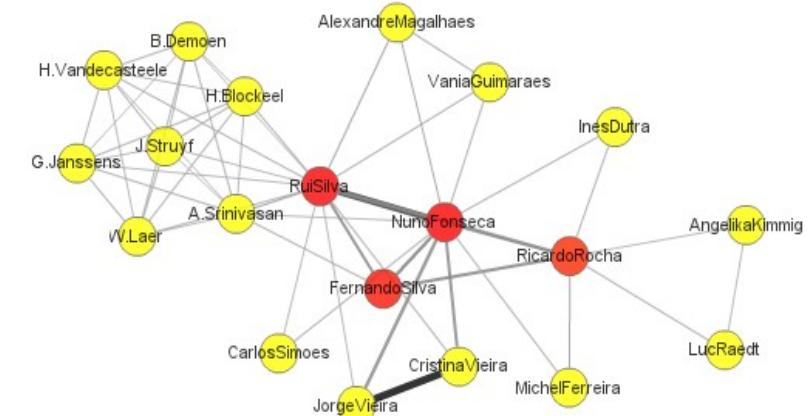
Brain

Complex Networks are Ubiquitous

Social

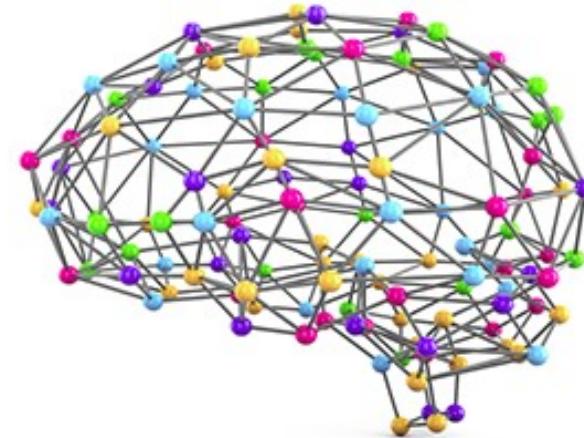


Facebook

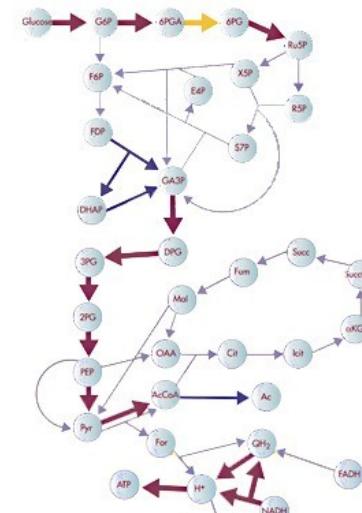


Co-authorship

Biological



Brain



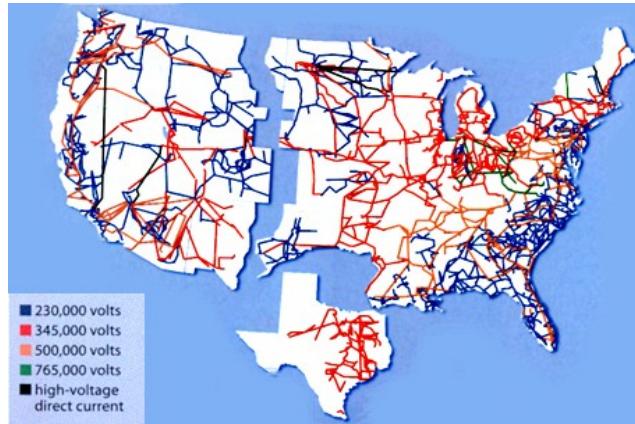
Metabolism
(proteins)

Complex Networks are Ubiquitous

Spatial

Complex Networks are Ubiquitous

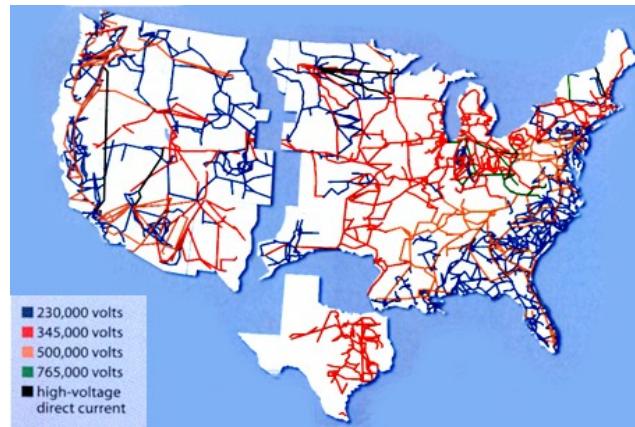
Spatial



Power

Complex Networks are Ubiquitous

Spatial

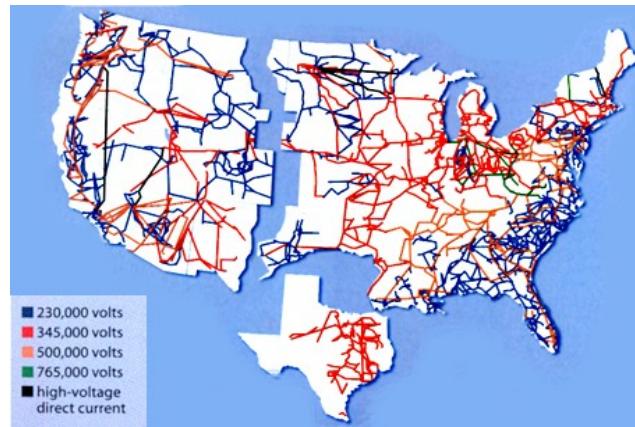


Power



Complex Networks are Ubiquitous

Spatial



Power

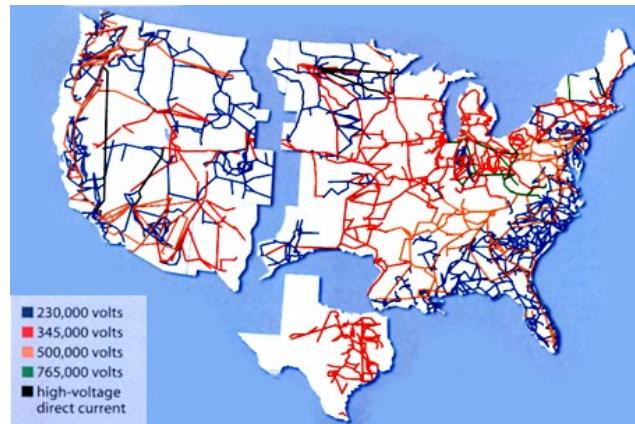


Roads

Software

Complex Networks are Ubiquitous

Spatial

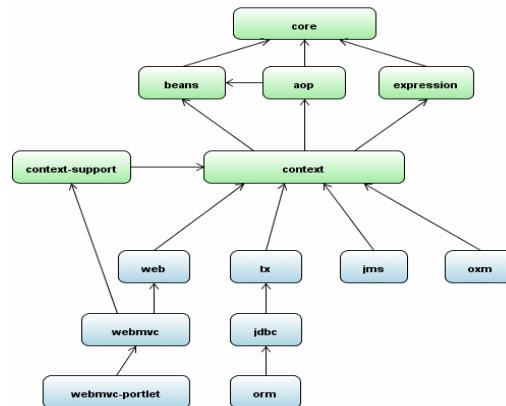


Power



Roads

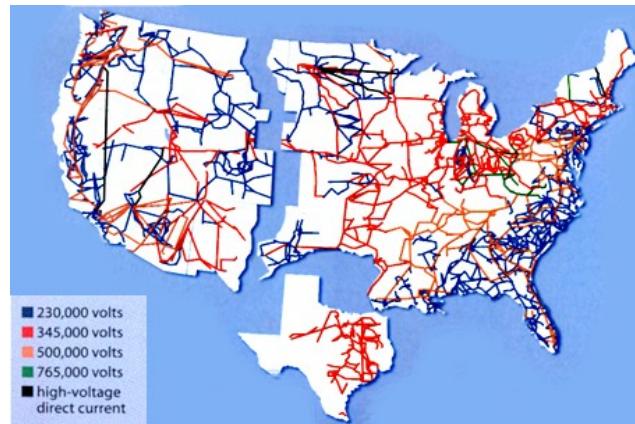
Software



Module Dependency

Complex Networks are Ubiquitous

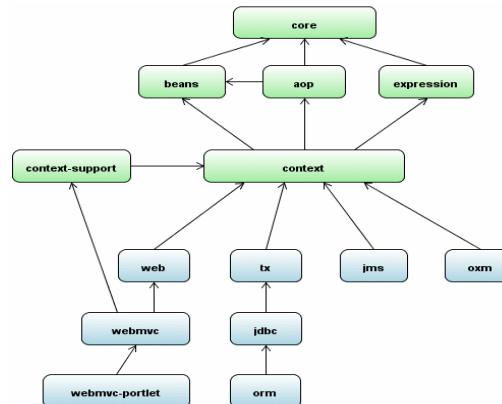
Spatial



Power

Roads

Software

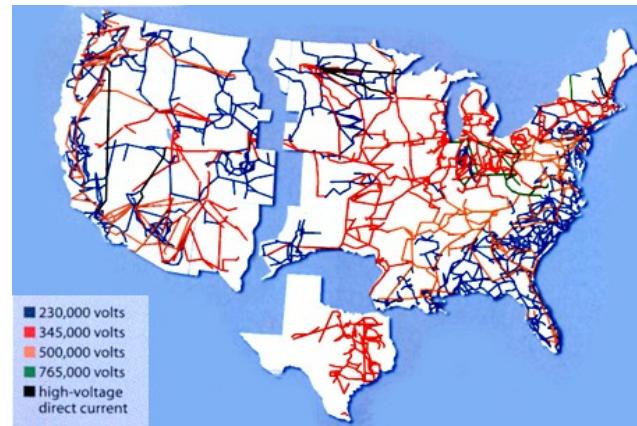


Module Dependency

Text

Complex Networks are Ubiquitous

Spatial

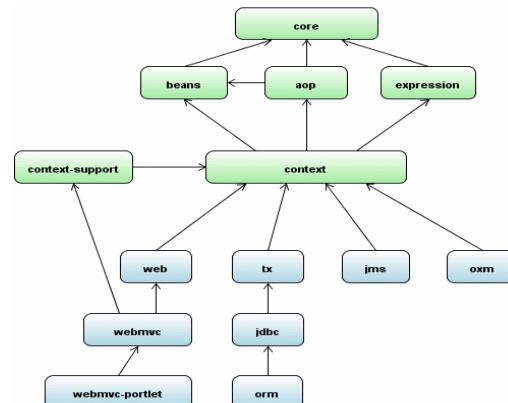


Power



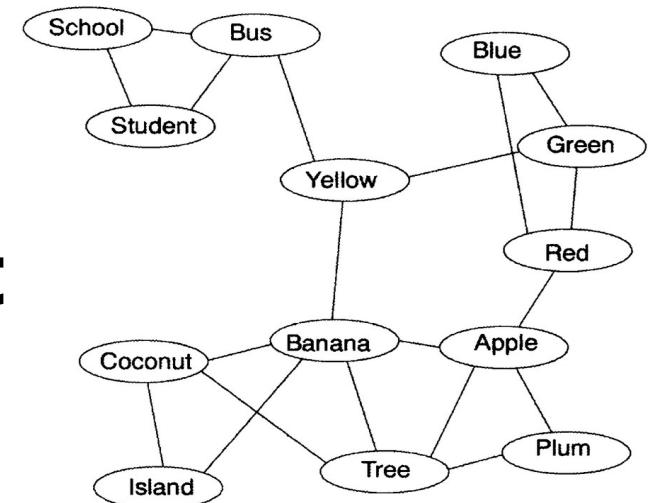
Roads

Software



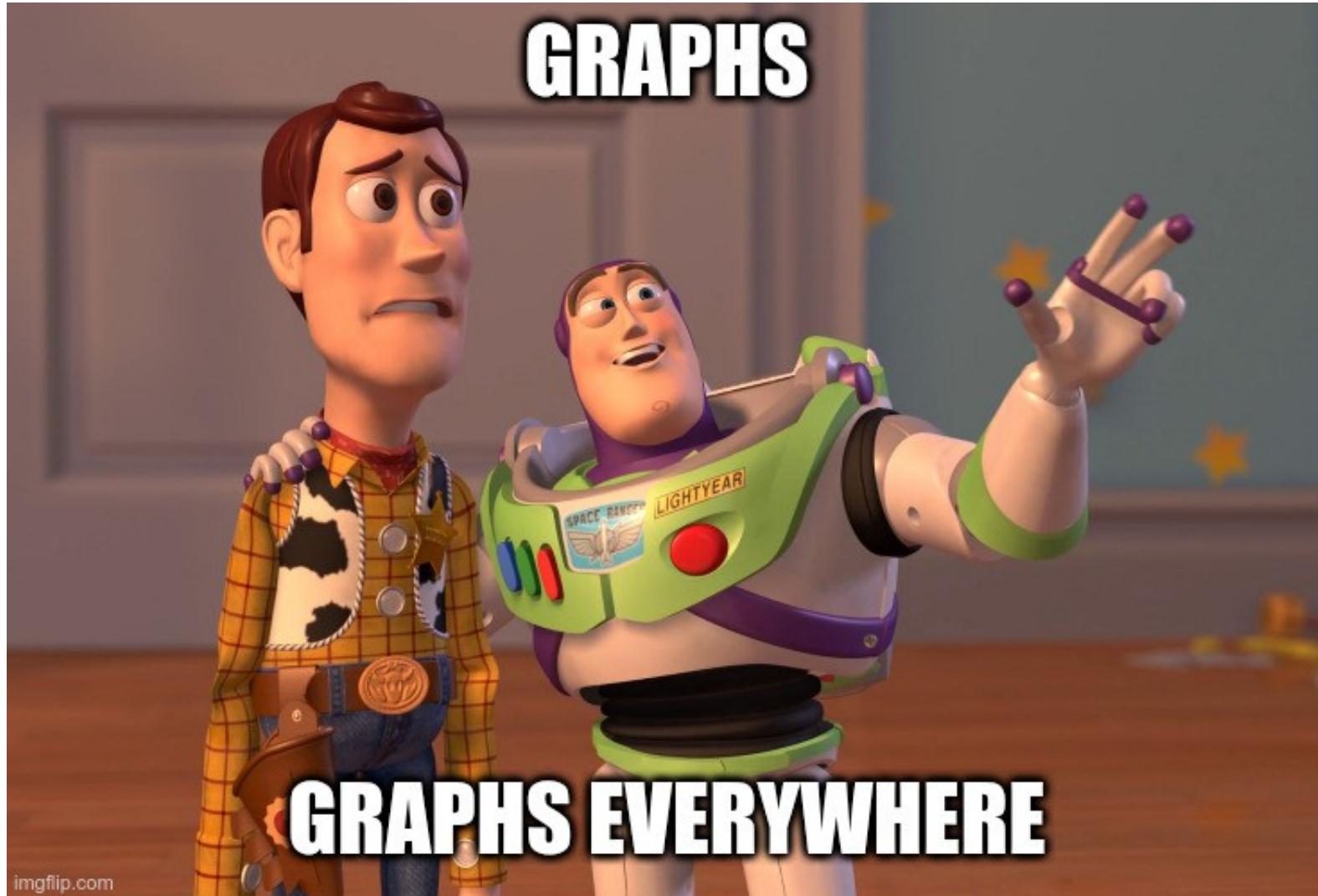
Module
Dependency

Text



Semantic

Complex Networks are Ubiquitous



Network Science

Behind many complex systems there is a **network** that defines the **interactions** between the components

In order to understand the systems...
we need to understand the **networks!**

Network Science

- **Network Science** has been emerging on this century as a new discipline:
 - Origins on **graph theory** and **social** network research

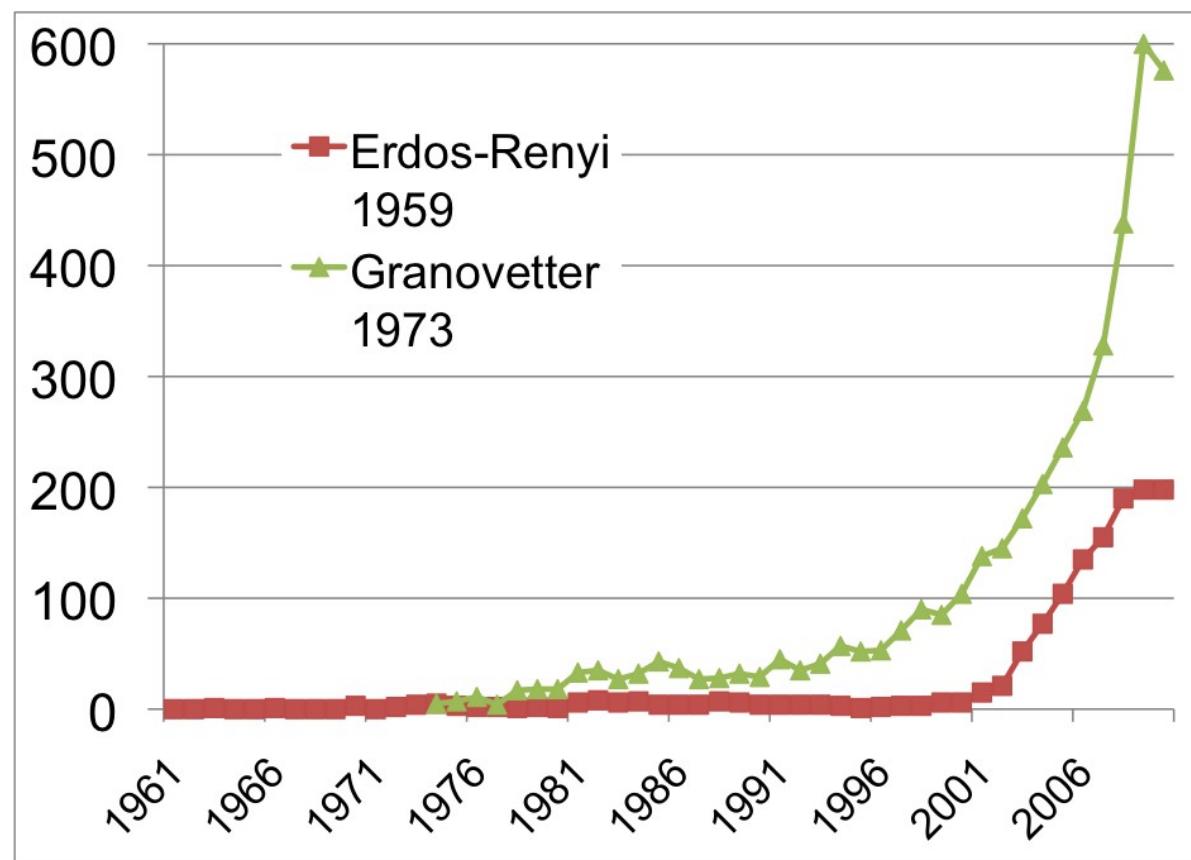


Image: Adapted from (Barabasi, 2015)

Why now?

- Two main contributing factors:

Why now?

- Two main contributing factors:

1) The emergence of **network maps**

Why now?

- Two main contributing factors:

1) The emergence of **network maps**

- Movie actor network: 1998
- World Wide Web: 1999
- Citation Network: 1998
- Metabolic Network: 2000
- PPI Network: 2001

Why now?

- Two main contributing factors:

1) The emergence of **network maps**

- Movie actor network: 1998
- World Wide Web: 1999
- Citation Network: 1998
- Metabolic Network: 2000
- PPI Network: 2001

- **436 nodes** – 2003
(email exchange, Adamic-Adar, SocNets)
- **43,553 nodes** – 2006
(email exchange, Kossinets-Watts, Science)
- **4.4 million nodes** – 2005
(friendships, Liben-Nowell, PNAS)
- **800 million nodes** – 2011
(Facebook, Backstrom et al.)

Size matters!

Why now?

- Two main contributing factors:

2) Universality of network characteristics

Why now?

- Two main contributing factors:

2) Universality of network **characteristics**

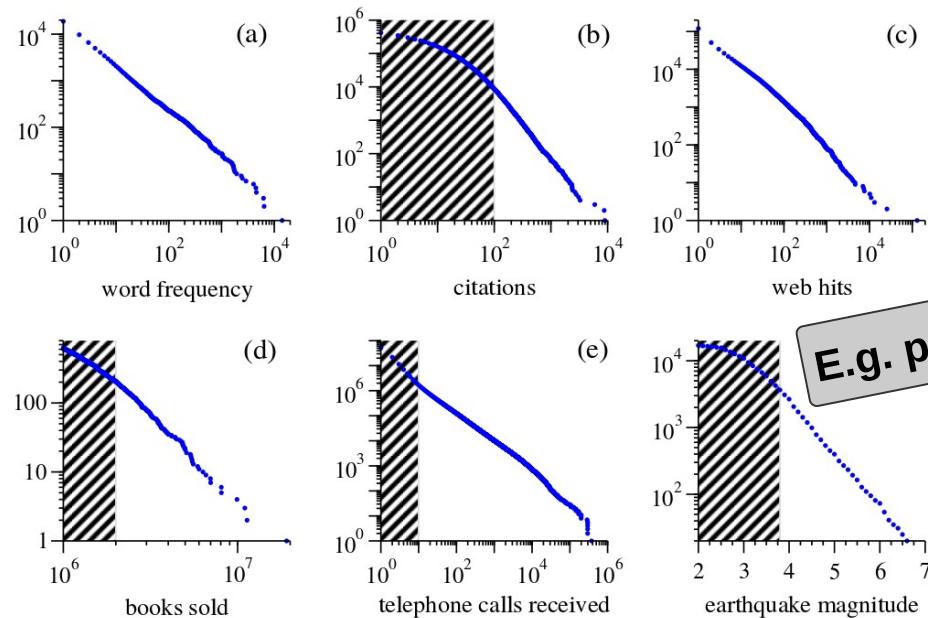
- The architecture and topology of networks from different domains exhibit more similarities than what one would expect

Why now?

- Two main contributing factors:

2) Universality of network characteristics

- The architecture and topology of networks from different domains exhibit more similarities than what one would



E.g. power laws

Many real world networks are power law

	exponent α (in/out degree)
film actors	2.3
telephone call graph	2.1
email networks	1.5/2.0
sexual contacts	3.2
WWW	2.3/2.7
internet	2.5
peer-to-peer	2.1
metabolic network	2.2
protein interactions	2.4

Image: Adapted from (Newman, 2005)

Image: Adapted from Leskovec, 2015

Impact of Network Science: Economic

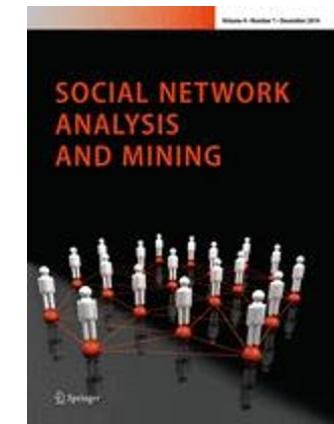
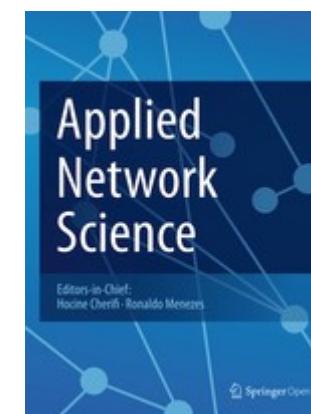
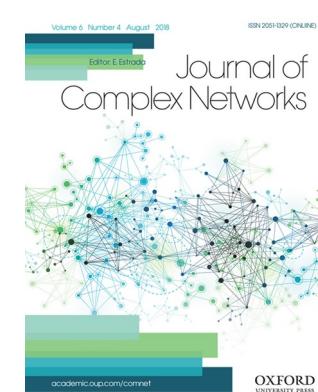
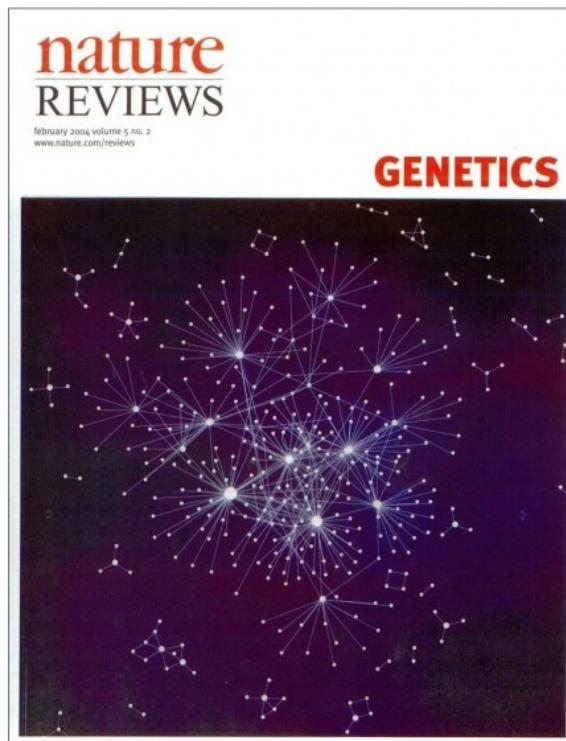


Google

facebook®

CISCO™

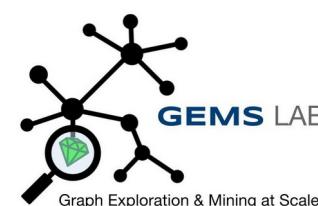
Impact of Network Science: Scientific



Northeastern University
Network Science Institute



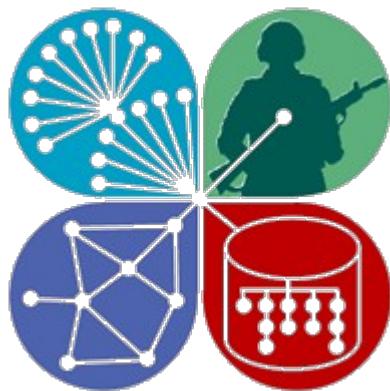
Indiana University
Network Science Institute



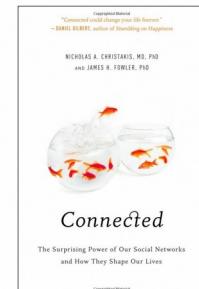
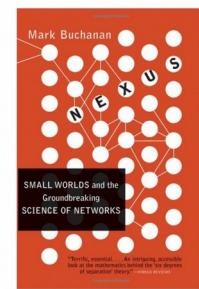
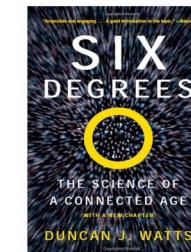
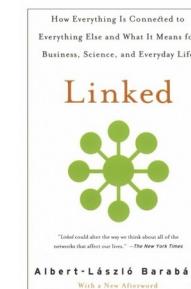
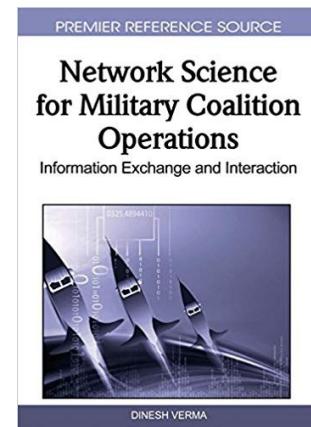
Winter School on
Network Science



Impact of Network Science: Societal



Network Science Center
West Point 



Reasoning about Networks

- **What do we hope to achieve from studying networks?**
 - Patterns and statistical **properties** of network data
 - **Design principles** and **models**
 - **Algorithms** and **predictive** models to answer questions and make predictions

Mining and Learning with Graphs

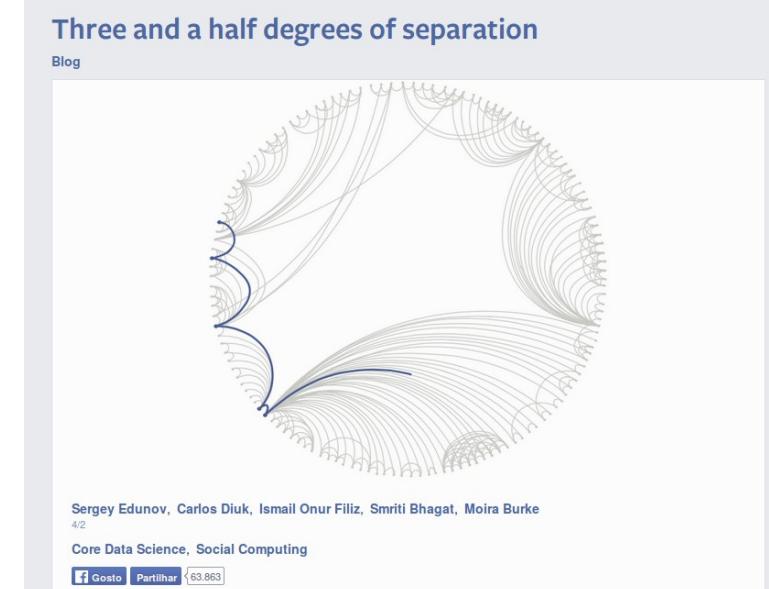
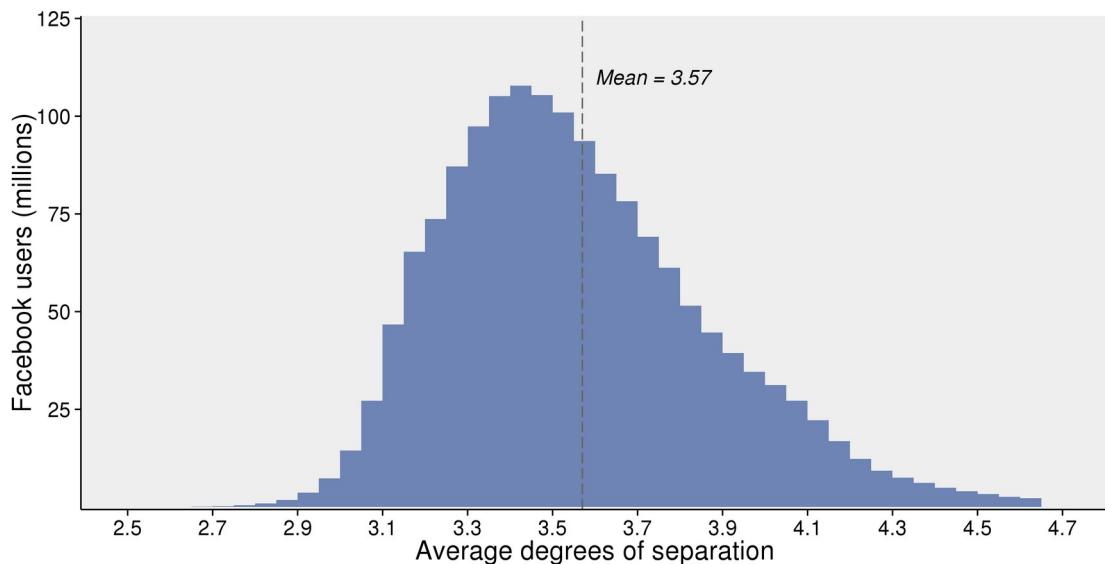
- **How do we mine networks?**
 - **Empirically:** Study network data to find organizational principles
 - How do we measure and quantify networks?
 - **Mathematical models:** Graph theory and statistical models
 - Models allow us to understand behaviors and distinguish surprising from expected phenomena
 - **Algorithms** for analyzing graphs
 - Hard computational challenges

Network Science Topics

- Some possible tasks:

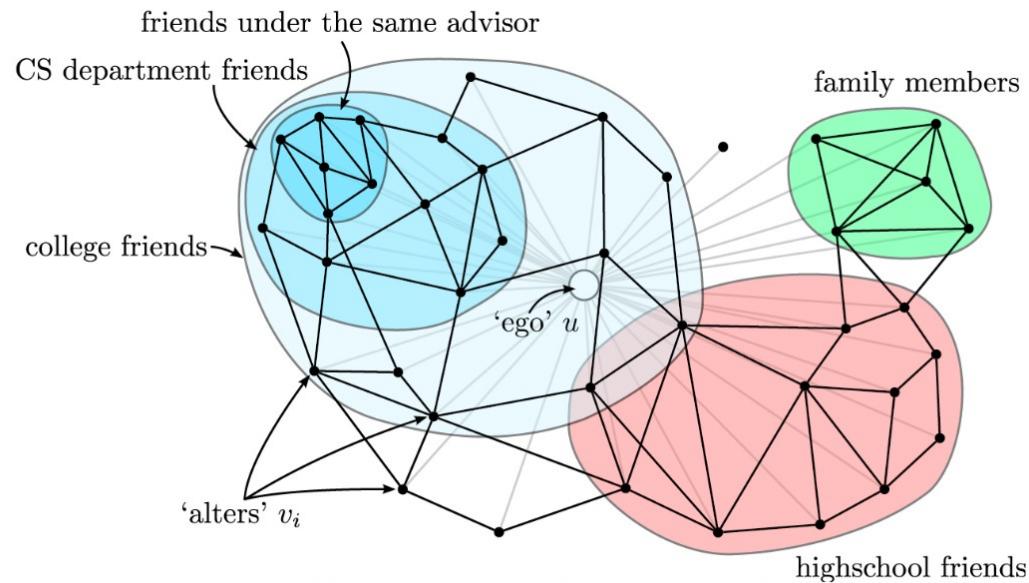
Network Science Topics

- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”



Network Science Topics

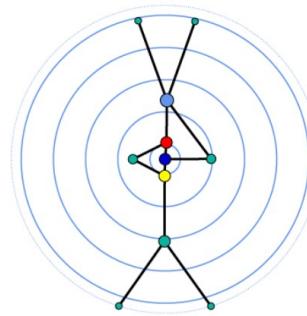
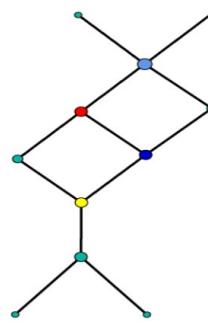
- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”
 - Community Detection
 - What groups of nodes are “related”?



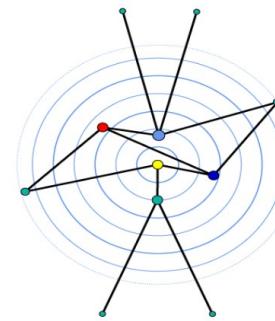
Discover circles and why they exist

Network Science Topics

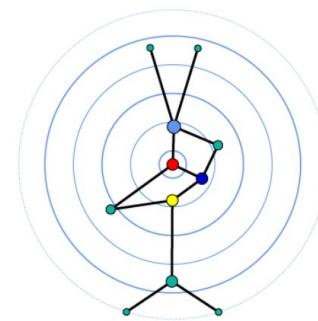
- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”
 - Community Detection
 - What groups of nodes are “related”?
 - Node Classification
 - Importance and function of a certain node?



Closeness



Betweenness



Eigenvector

Network Science Topics

- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”
 - Community Detection
 - What groups of nodes are “related”?
 - Node Classification
 - Importance and function of a certain node?
 - Network Comparison
 - What is the type of the network?

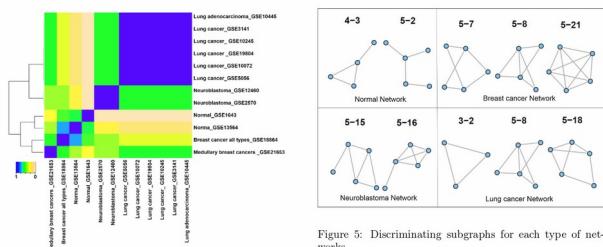
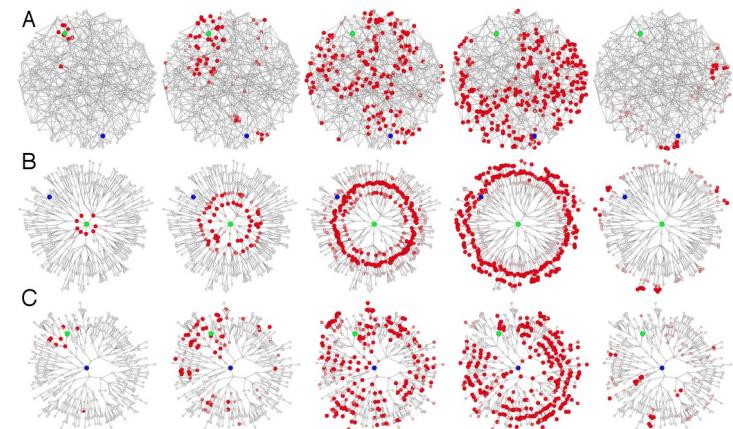


Figure 5: Discriminating subgraphs for each type of networks.

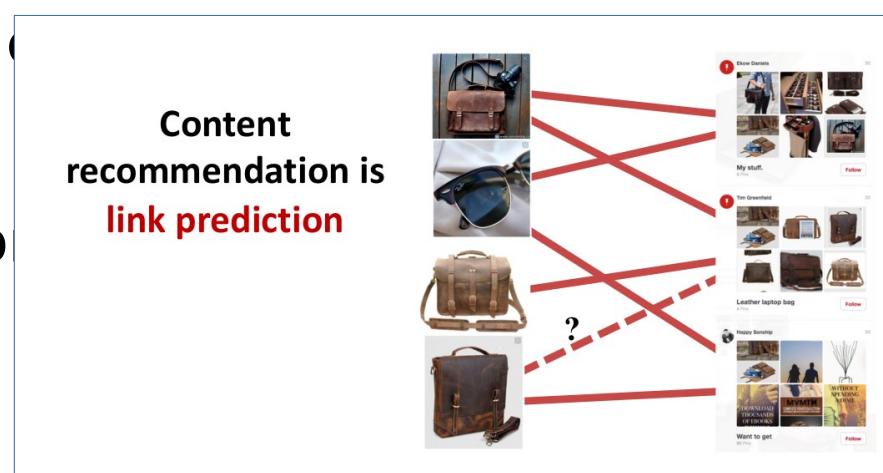
Network Science Topics

- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”
 - Community Detection
 - What groups of nodes are “related”?
 - Node Classification
 - Importance and function of a certain node?
 - Network Comparison
 - What is the type of the network?
 - Information Propagation
 - Epidemics? Robustness?



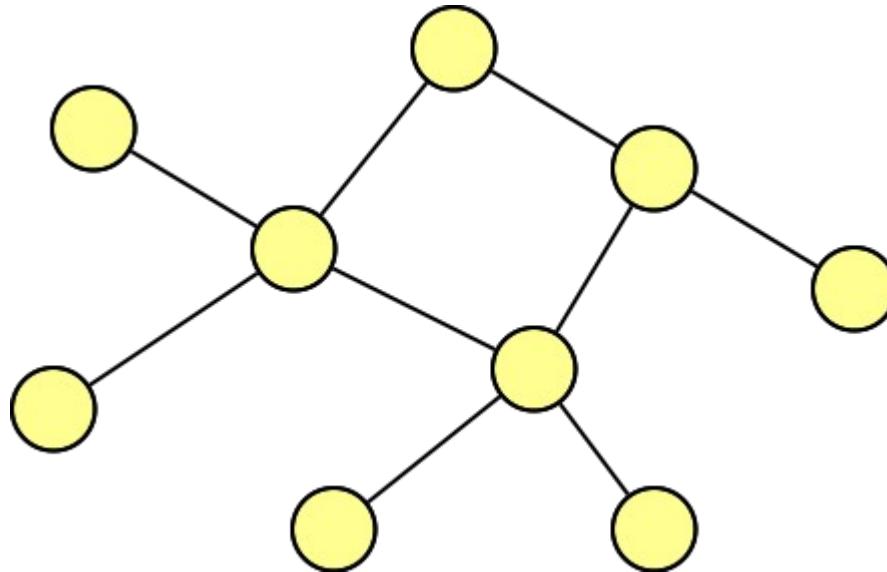
Network Science Topics

- Some possible tasks:
 - General Patterns
 - Ex: “scale-free”, “small-world”
 - Community Detection
 - What groups of nodes are “related”?
 - Node Classification
 - Importance and function of a node
 - Network Comparison
 - What is the type of the network?
 - Information Propagation
 - Epidemics? Robustness?
 - Link prediction
 - Future connections? Errors in graph constructions?



Brief Introduction to Graph Theory and Network Vocabulary

Terminology



- **Objects:** nodes, vertices V
- **Interactions:** links, edges E
- **System:** network, graph $G(V,E)$

Networks or Graphs?

- **Network** often refers to real systems
 - Web, Social network, Metabolic network
 - Language: Network, node, link
- **Graph** is a mathematical representation of a network
 - Web graph, Social graph (a Facebook term)
 - Language: Graph, vertex, edge

We will try to make this distinction whenever it is appropriate, but
in most cases we will use the two terms interchangeably

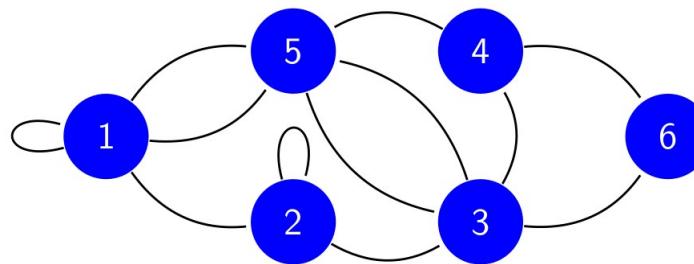
Choosing the Network

- If you connect individuals that work with each other, you will explore a **professional network**
- If you connect those that are friends, you will be exploring a **friendship network**
- If you connect scientific papers that cite each other, you will be studying the **citation network**
- **Another example:** if you connect all papers with the same word in the title, what will you be exploring?
- There might be **several possible representations**

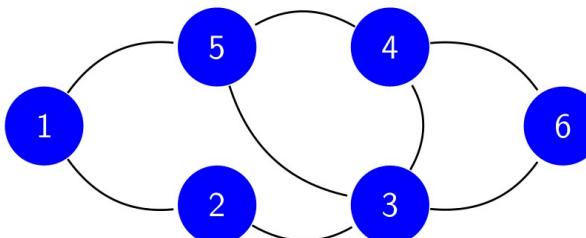
The choice of the network representation of a given domain determines our ability to use it successfully

Simple and multi-graphs

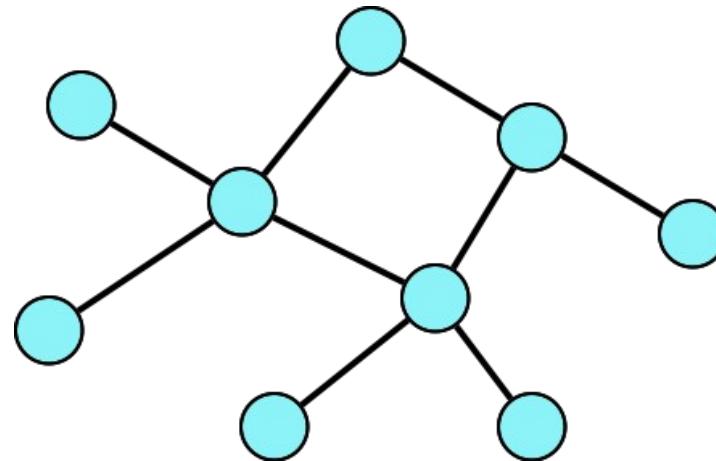
- In general, graphs may have self-loops and multi-edges
 - A graph with either is called a **multi-graph**



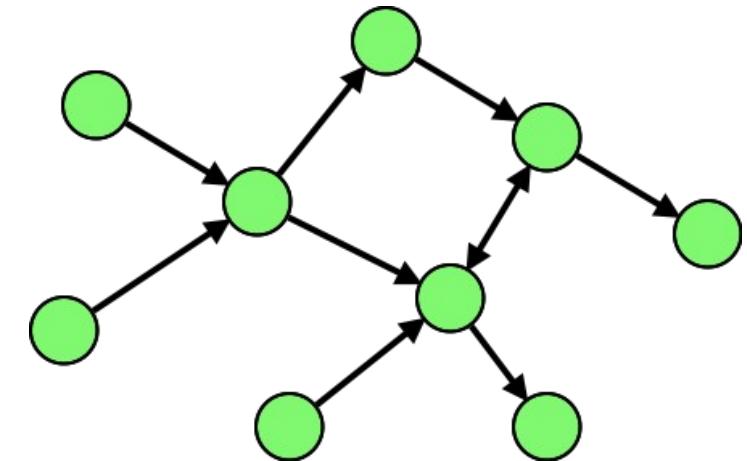
- We will mostly work with **simple graphs**, with no self-loops or multi-edges



Network Types



Undirected



Directed

- co-authorship networks
- actor networks
- facebook friendships
- www hyperlinks
- phone calls
- roads network

Network Types

Edge Attributes

- Examples:
 - **Weight** (duration call, distance road, ...)
 - **Ranking** (best friend, second best friend, ...)
 - **Type** (friend, relative, co-worker, ...) [colored edges]
 - We can have a set of **multiple** attributes

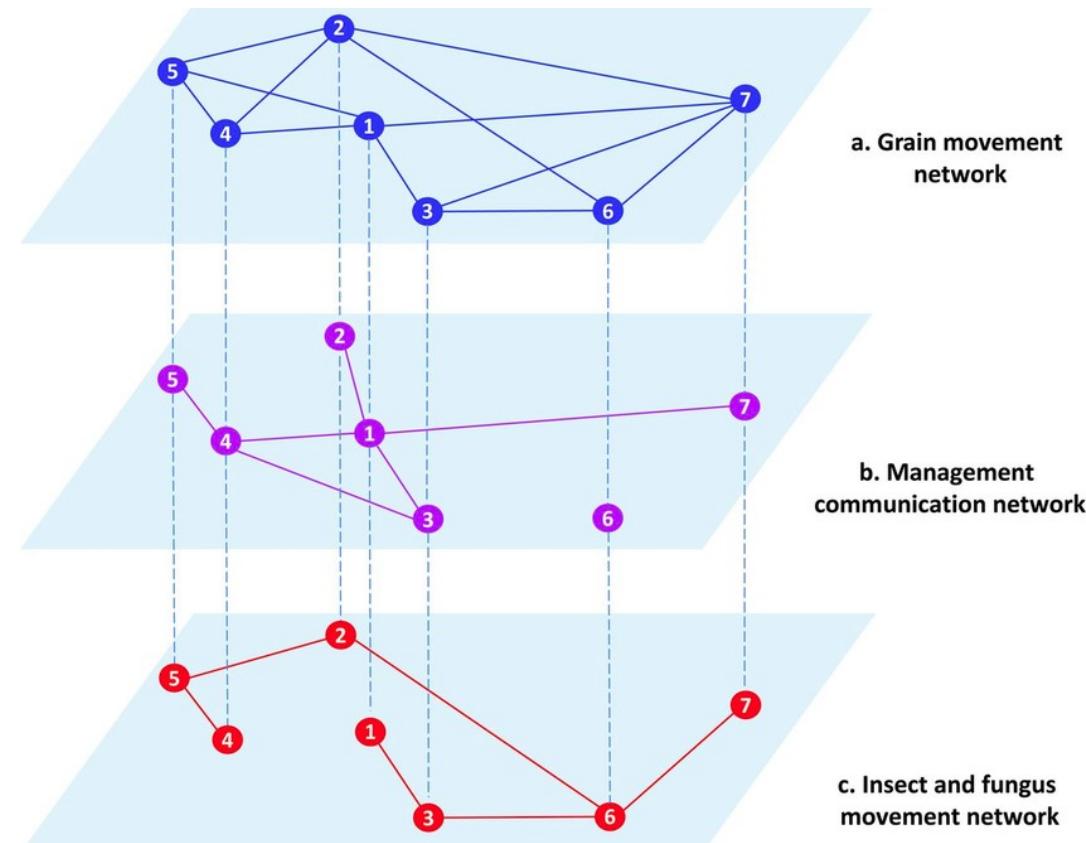
Node Attributes

- Examples:
 - **Type** (nationality, sex, age, ...) [colored nodes]
 - We can have a set of **multiple** attributes

Network Types

Multiplex Networks

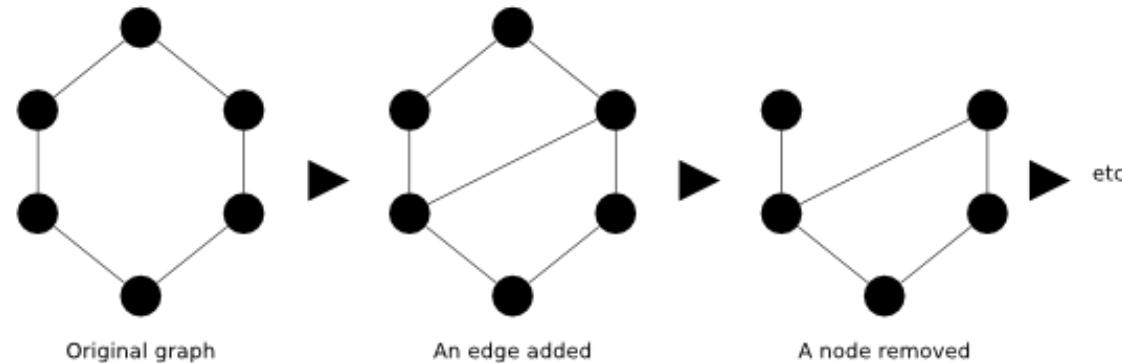
- Different layers (types) of connections



Network Types

Temporal Networks

- Evolution over time

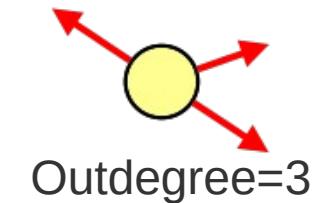


Node Properties

- **From immediate connections**

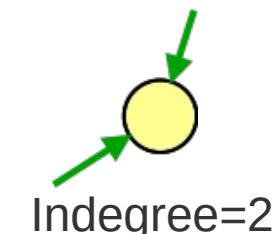
- **Outdegree**

how many directed edges originate at node



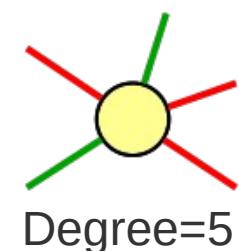
- **Indegree**

how many directed edges are incident on a node



- **Degree** (in or out)

number of outgoing and incoming edges



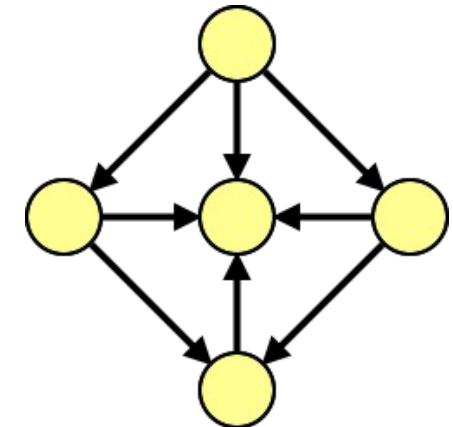
Node Properties

- **Degree related metrics:**

- **Degree sequence**

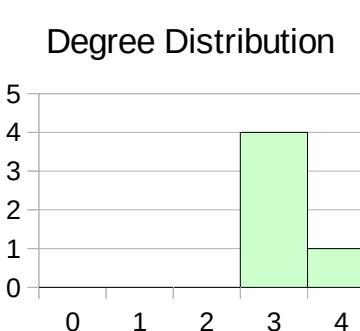
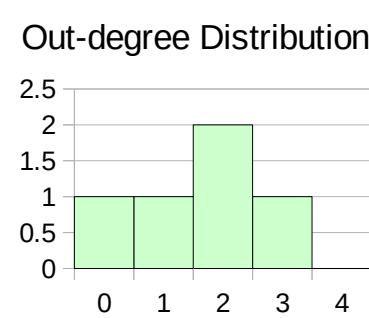
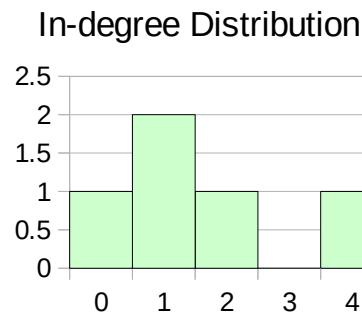
an ordered list of the (in,out) degree of each node

- In-degree sequence: [4, 2, 1 , 1, 0]
 - Out-degree sequence: [3, 2, 2, 1, 0]
 - Degree sequence: [4, 3, 3, 3, 3]



- **Degree Distribution**

a frequency count of the occurrences of each degree
[usually plotted as probability → normalization]



Sparsity of Networks

- Real Networks are usually very Sparse!

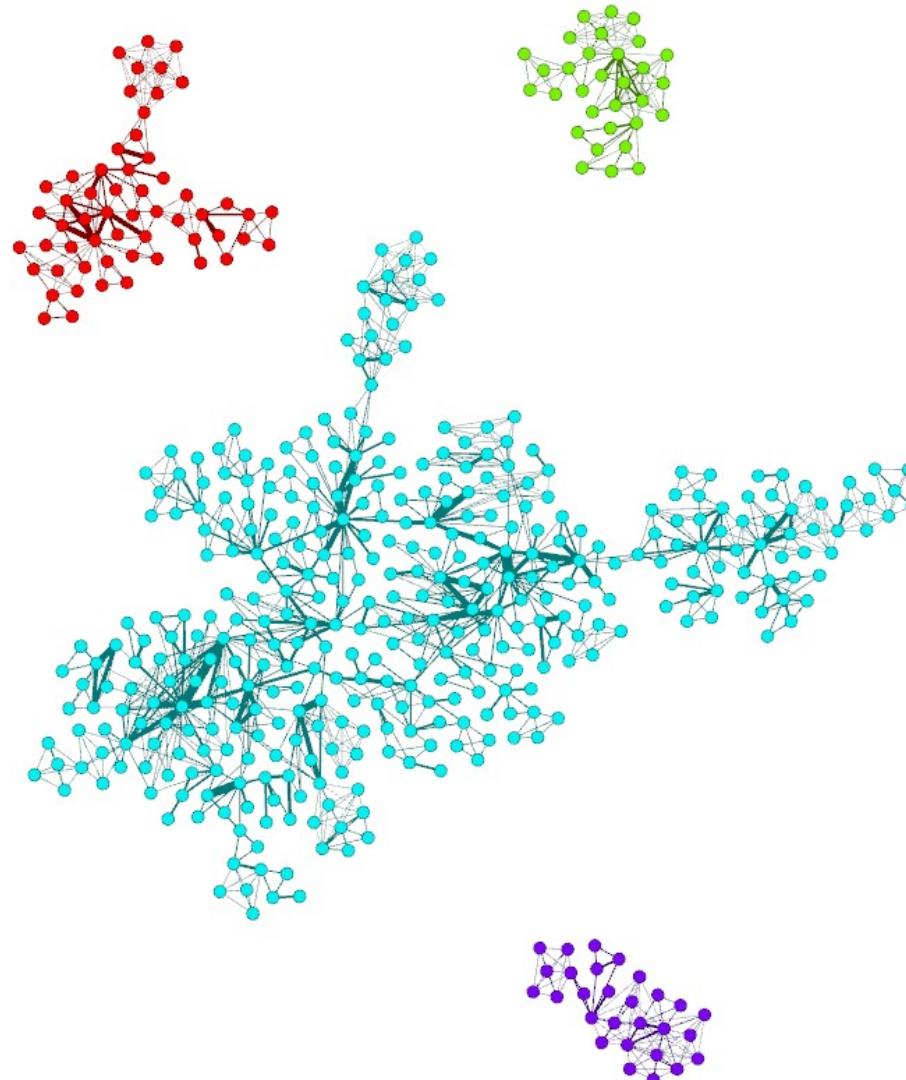
Network	Dir/Undir	Nodes	Edges	Avg. Degree
Internet	Undirected	192,244	609,066	6.33
WWW	Directed	325,729	1,479,134	4.60
Power Grid	Undirected	4,941	6,594	2.67
Mobile Phone Calls	Directed	36,595	91,826	2.51
Email	Directed	57,194	103,731	1.81
Science Collaboration	Undirected	23,133	93,439	8.08
Actor Network	Undirected	702,388	29,397,908	83.71
Citation Network	Directed	449,673	4,689,479	10.43
E. Coli Metabolism	Directed	1,039	5,082	5.58
Protein Interactions	Undirected	2,018	2,930	2.90

- A graph where every pair of nodes is connected is called a **complete graph** (or a **clique**)

Table: Adapted from (Barabasi, 2015)

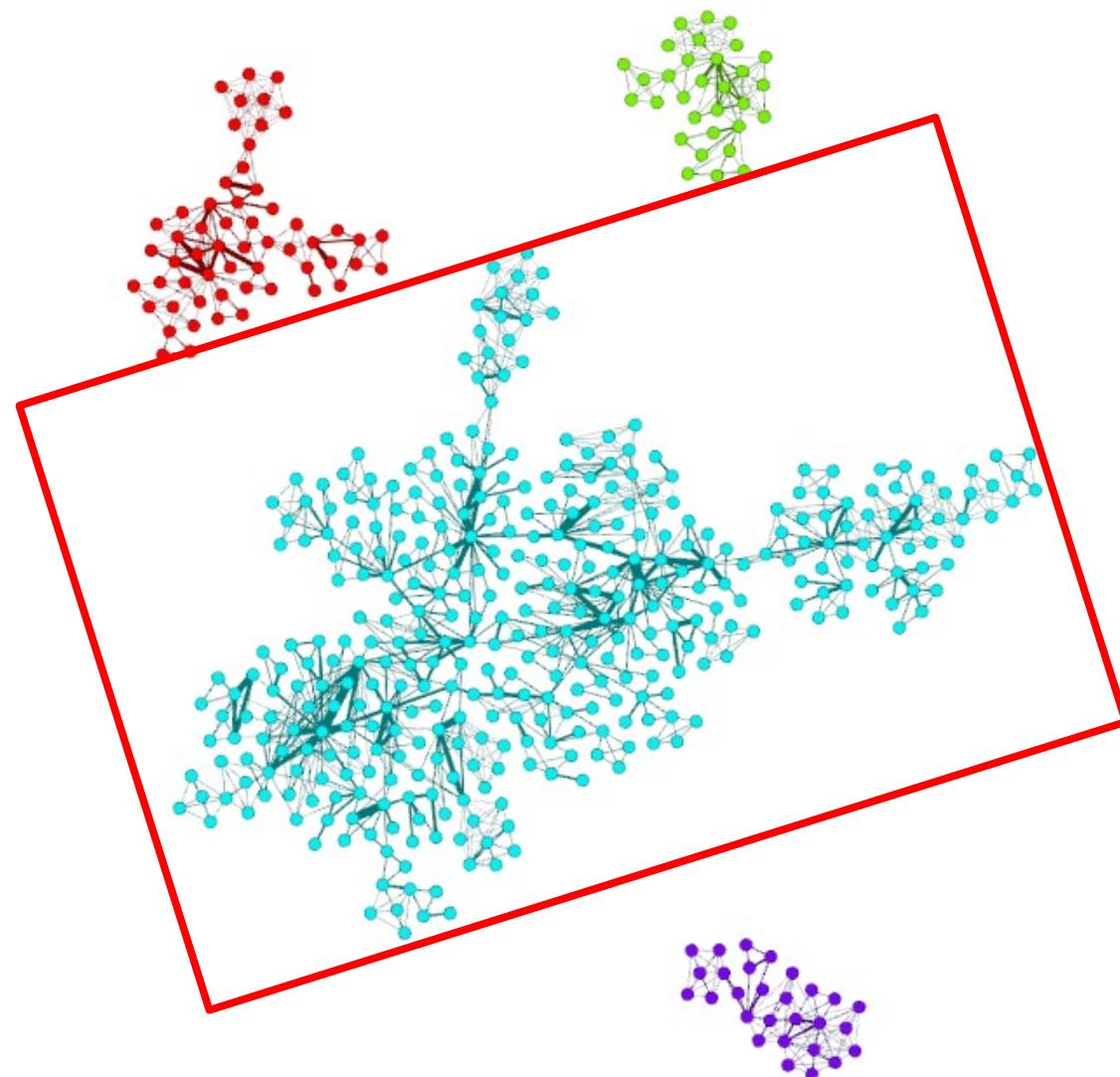
Connectivity

- Not everything is connected



Connectivity

- If the largest component has a large fraction of the nodes we call it the **giant component**



Bipartite

- A **bipartite graph** is a graph whose nodes can be divided into two disjoint sets **U** and **V** such that every edge connects a node in **U** to one in **V**.

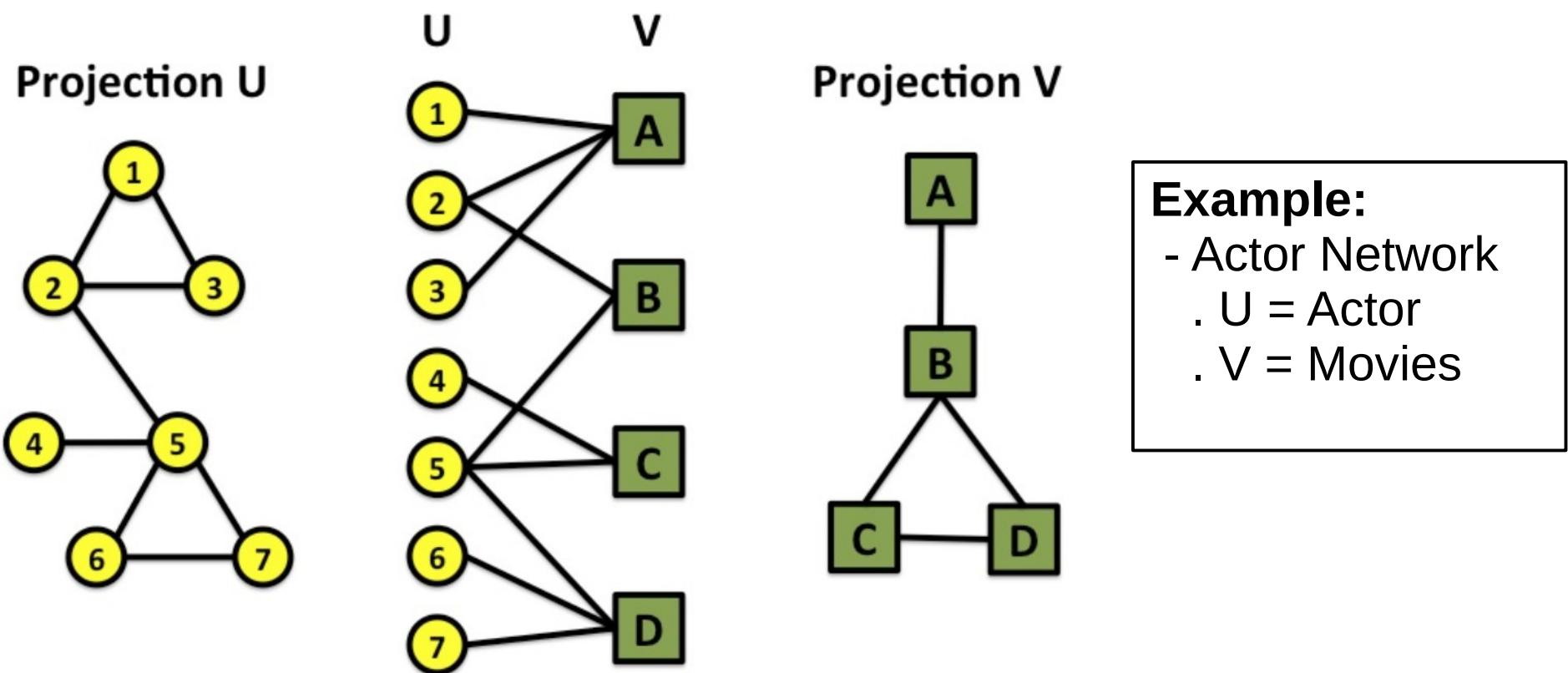
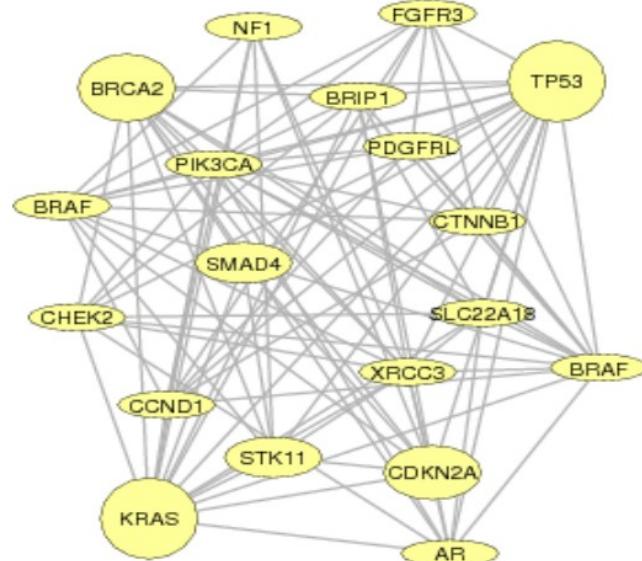
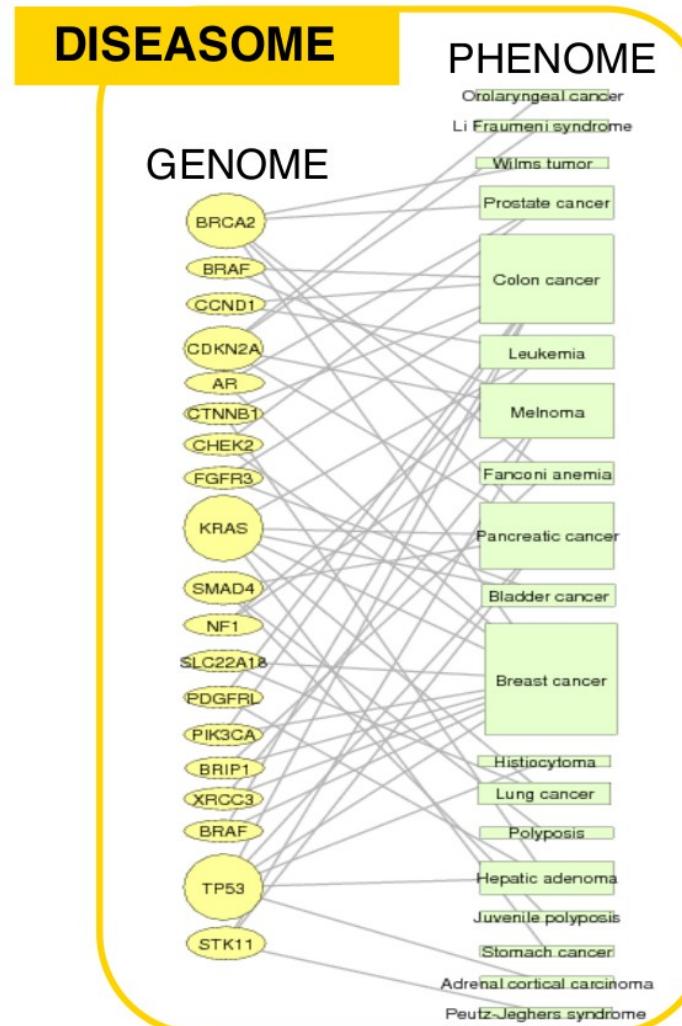


Image: Adapted from Leskovec, 2015

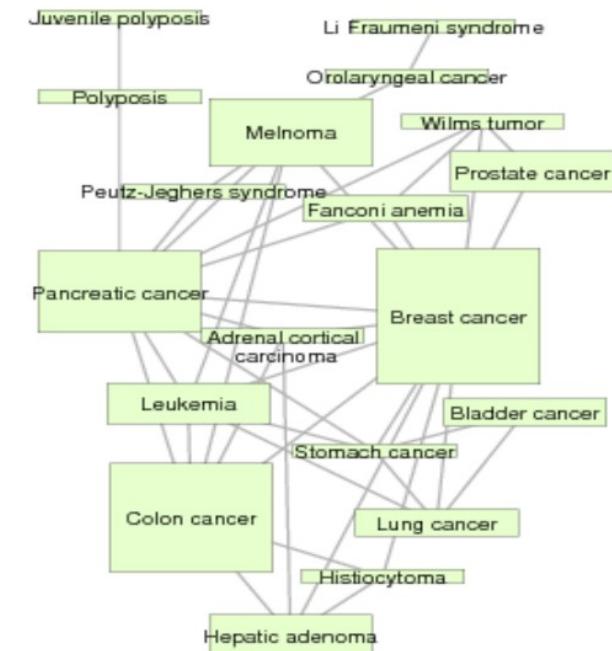
Bipartite Network Projections



Gene network



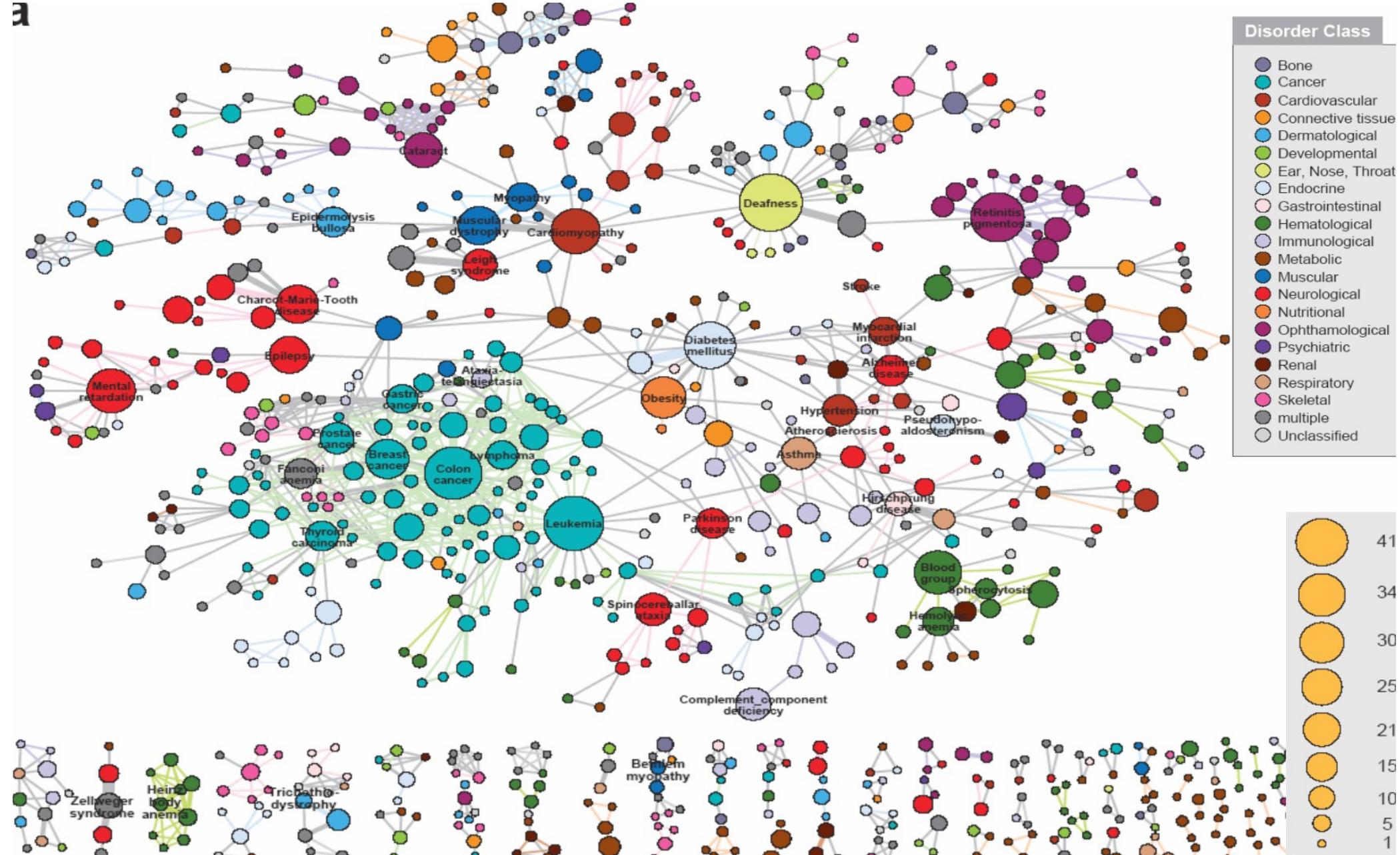
Goh, Cusick, Valle, Childs, Vidal & Barabási, PNAS (2007)



Disease network

Bipartite - Human Disease Network

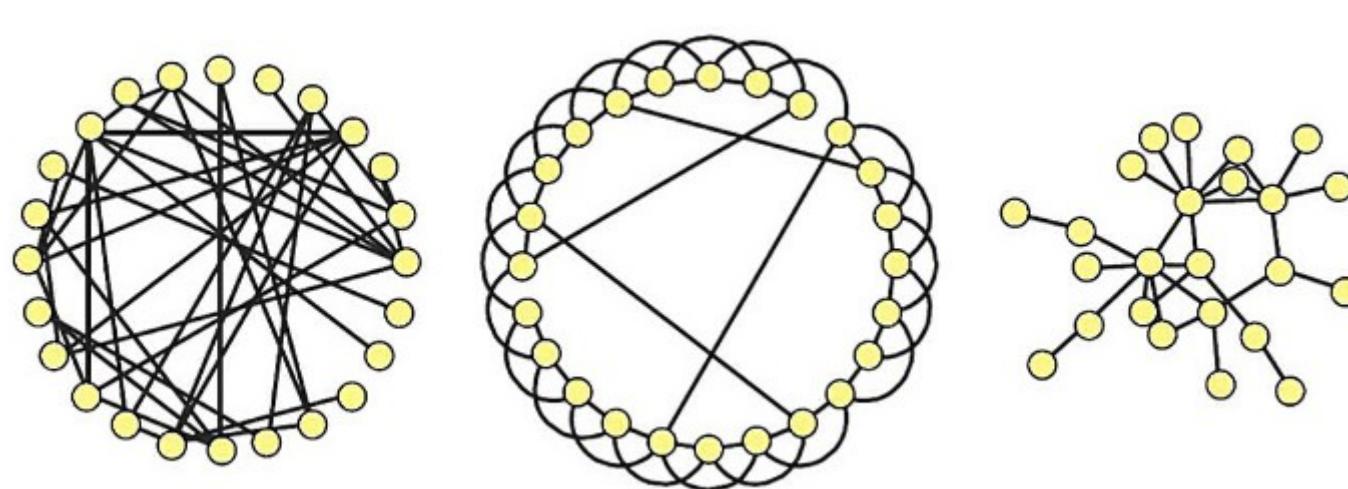
a



Measuring Networks and Random Graph Models



Pedro Ribeiro
(DCC/FCUP & CRACS/INESC-TEC)



(Heavily based on slides from Jure Leskovec and Lada Adamic@ Stanford University - CS224W)

Network Properties: how to measure a network?

Plan: Key Network Properties

- (1) Degree distribution $P(k)$
- (2) Path Length h
- (3) Clustering coefficient C
- (4) Connected components s

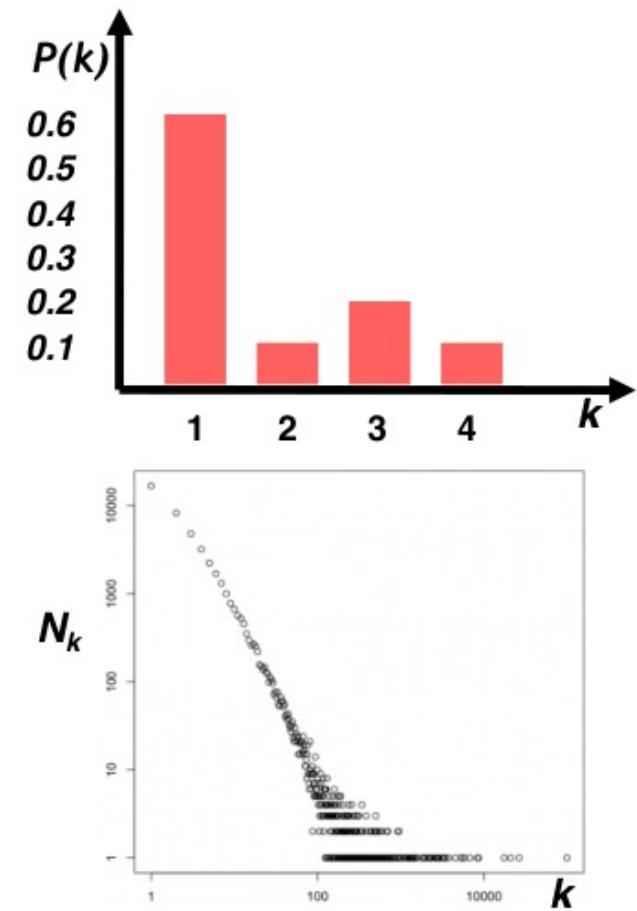
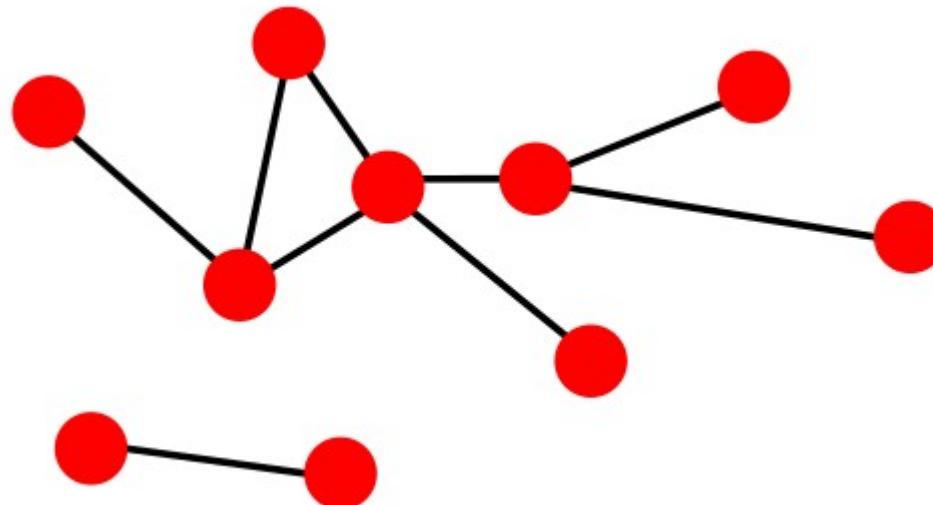
(1) Degree Distribution

- Degree distribution $P(k)$: probability that a randomly chosen node has degree k

$$N_k = \# \text{ nodes with degree } k$$

- Normalized histogram:

$$P(k) = N_k / N \rightarrow \text{plot}$$



(2) Paths in a Graph

- A **walk** is a sequence of nodes in which each node is linked to the next one

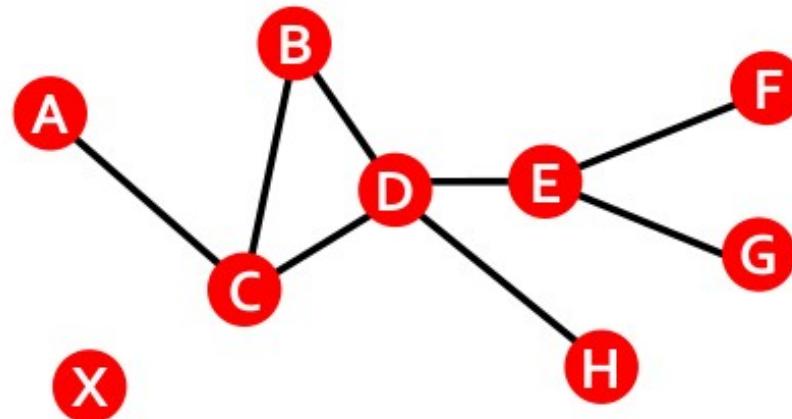
$$P_n = \{i_0, i_1, i_2, \dots, i_n\} \quad \text{or}$$

$$P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n)\}$$

- A **trail** is a walk without repeated edges
- A **path** is a walk without repeated vertices

- Examples:

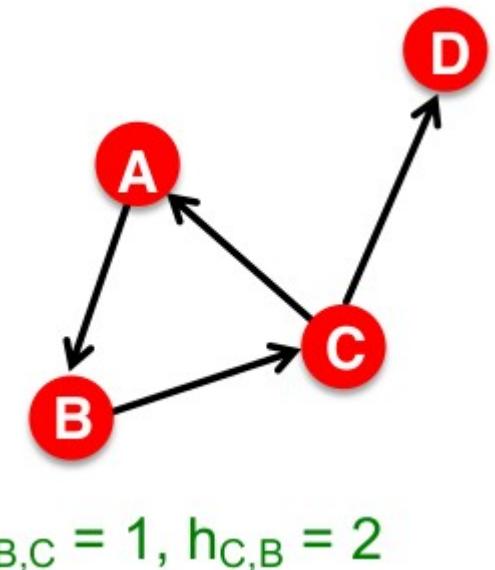
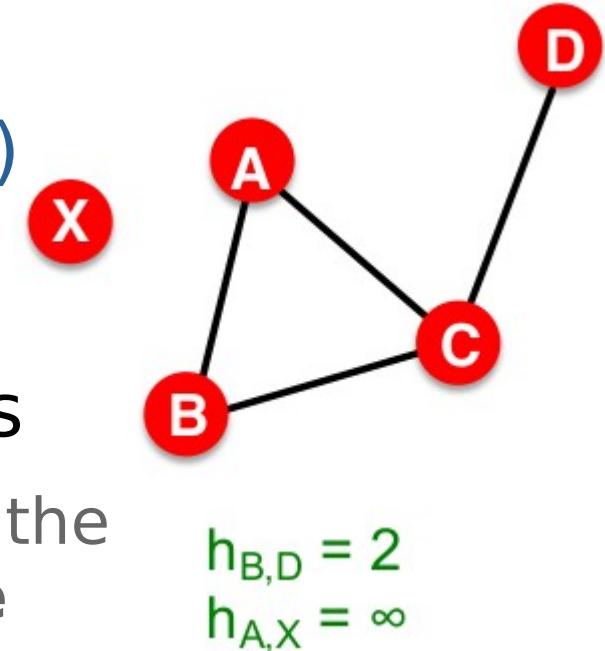
- Walk: ACBDCDEG
- Trail: ACBDC
- Path: ACDEF



- In a directed graph, a walk/trail/path can only follow the direction of the “arrow”

Distance in a Graph

- **Distance** (shortest path, geodesic) between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes
 - If the two nodes are **not connected**, the distance is usually defined as **infinite**
- In **directed graphs** paths need to follow the direction of the arrows
 - Consequence: distance is **not symmetric**: $h_{B,C} \neq h_{C,B}$



Network Diameter

- **Diameter:** The maximum (shortest path) distance between any pair of nodes in a graph
- **Average path length** for a connected graph (component) or a strongly connected (component of a) directed graph

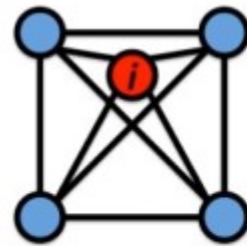
$$\bar{h} = \frac{1}{2E_{\max}} \sum_{i,j \neq i} h_{ij}$$

Where h_{ij} is the distance from node i to node j
 E_{\max} is max number of edges (total number of node pairs) = $n(n-1)/2$

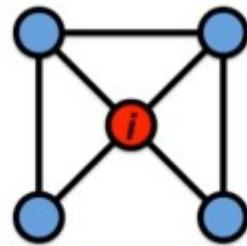
- Many times we compute the average only over the connected pairs of nodes (that is, we ignore “infinite” length paths)

(3) Clustering Coefficient

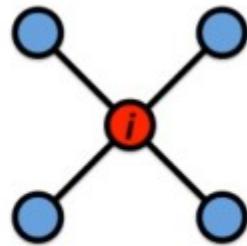
- **Clustering coefficient:**
 - What portion of i 's neighbors are connected?
 - Node i with degree k_i
 - $C_i \in [0,1]$
 - $C_i = \frac{2e_i}{k_i(k_i-1)}$ where e_i is the number of edges between the neighbors of node i



$$C_i = 1$$



$$C_i = 1/2$$

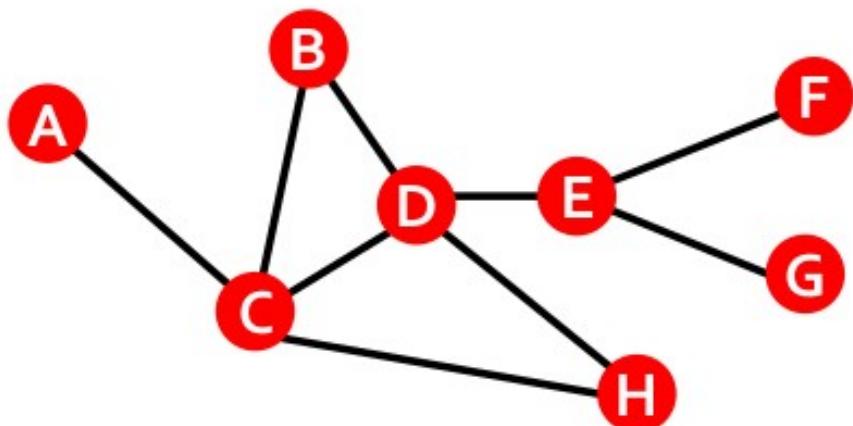


$$C_i = 0$$

- **Average clustering coefficient:** $C = \frac{1}{N} \sum_i^n C_i$

Clustering Coefficient

- **Clustering coefficient:**
 - What portion of i 's neighbors are connected?
 - Node i with degree k_i
 - $C_i = \frac{2e_i}{k_i(k_i-1)}$ where e_i is the number of edges between the neighbors of node i



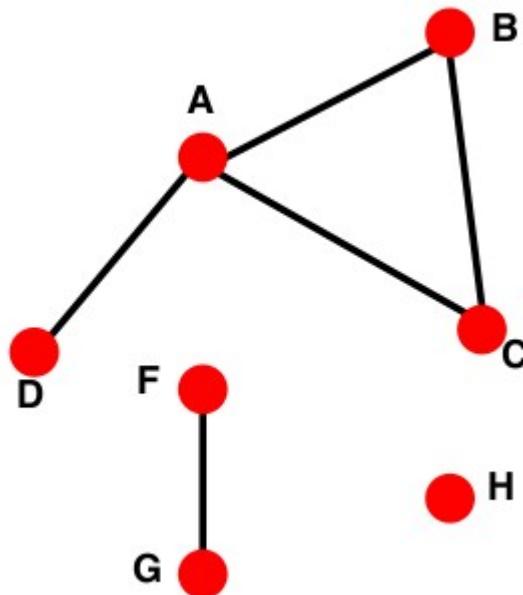
$$k_B=2, \quad e_B=1, \quad C_B = 2/2 = 1$$

$$k_D=4, \quad e_D=2, \quad C_D = 2/12 = 1/3$$

$$\text{Avg. Clustering: } C = 0.33$$

(4) Connectivity

- Size of the largest connected component
 - Largest set where any two vertices can be joined by a path
- **Largest component → Giant component**



How to find connected components:

- Start from random node and perform Breadth First Search (BFS)
- Label the nodes BFS visited
- If all nodes are visited, the network is connected
- Otherwise find an unvisited node and repeat BFS

Summary: Key Network Properties

- (1) Degree distribution $P(k)$
- (2) Path Length h
- (3) Clustering coefficient C
- (4) Connected components s

**Measuring these properties
in a Real World Graph**

MSN Messenger



- **MSN Messenger**
 - 1 month activity
 - 245 million users logged in
 - 180 million users engaged in conversations
 - More than 30 billion conversations
 - More than 255 billion exchanged messages

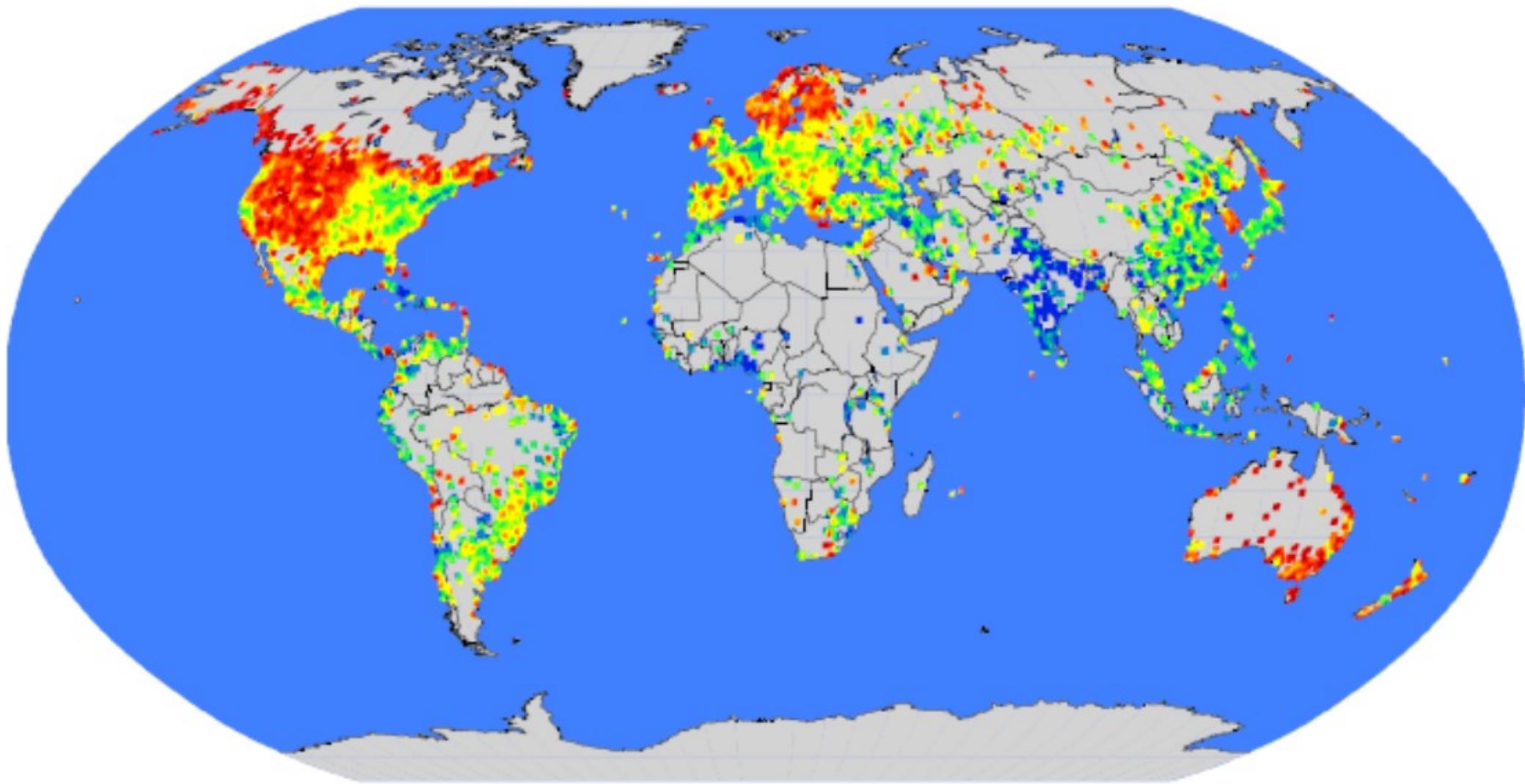
Planetary-Scale Views on a Large Instant-Messaging Network

WWW 2008

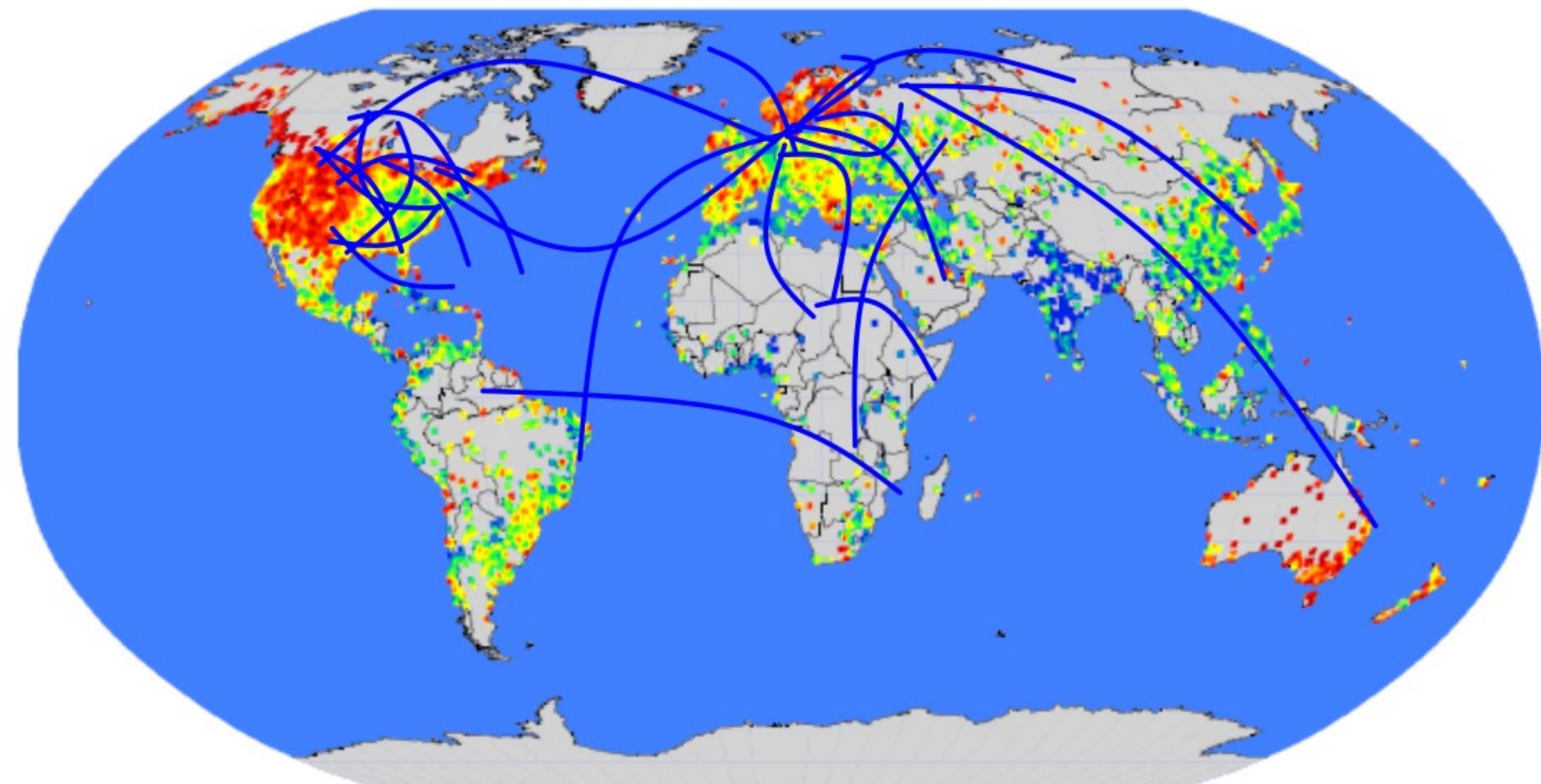
Jure Leskovec *
Carnegie Mellon University
jure@cs.cmu.edu

Eric Horvitz
Microsoft Research
horvitz@microsoft.com

Spatial Network: Geography

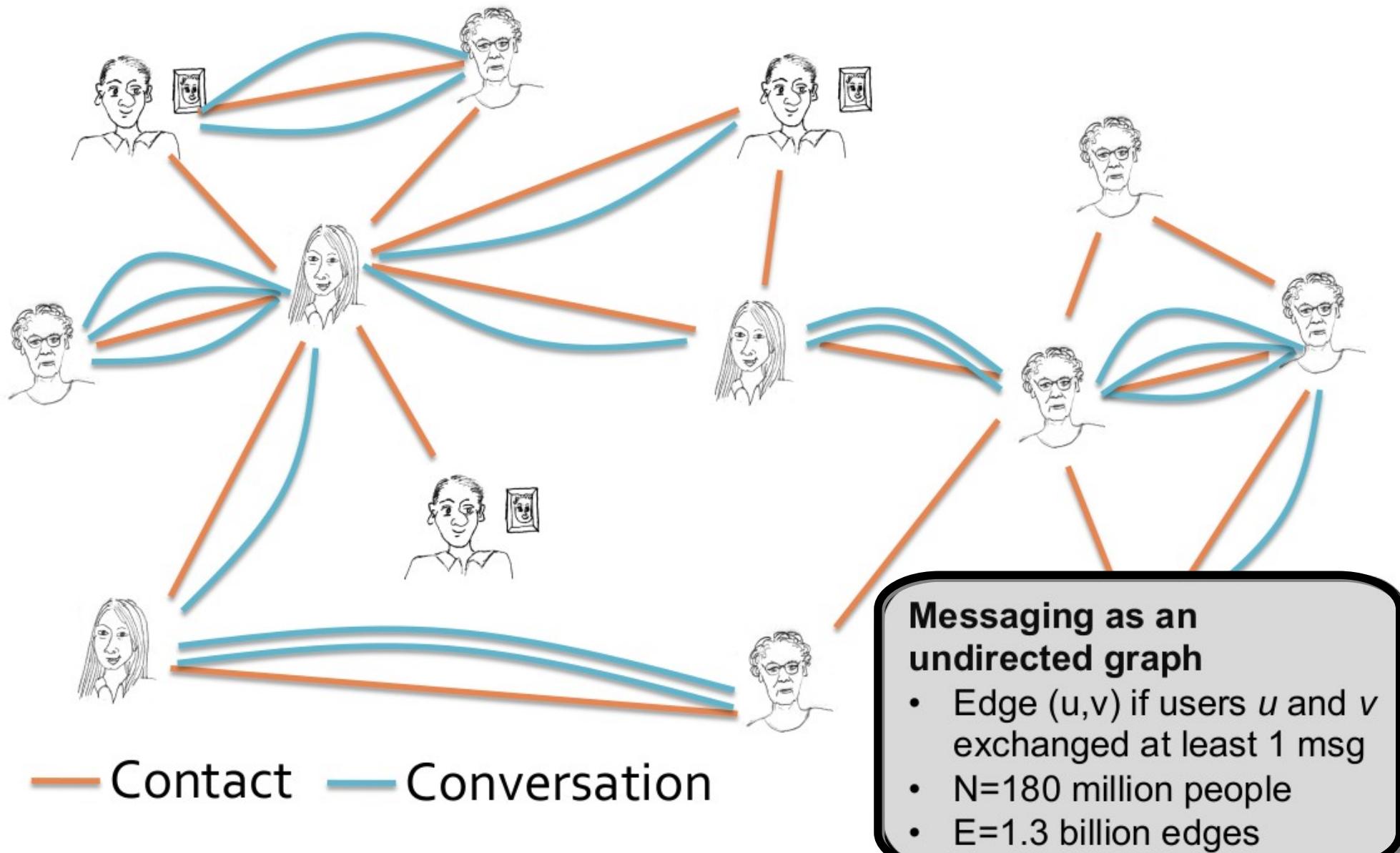


Communication → Connections

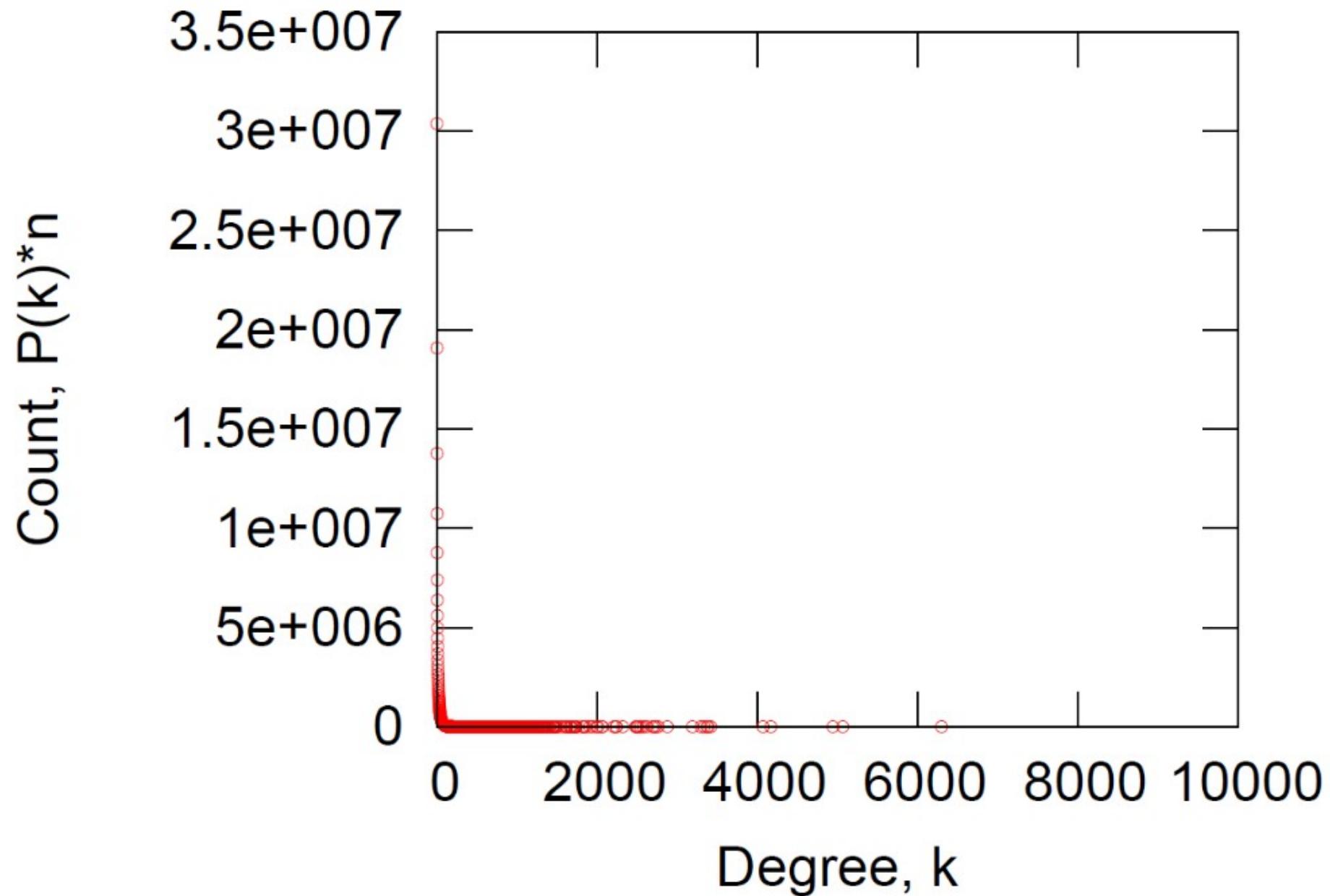


Network: 180M people, 1.3B edges

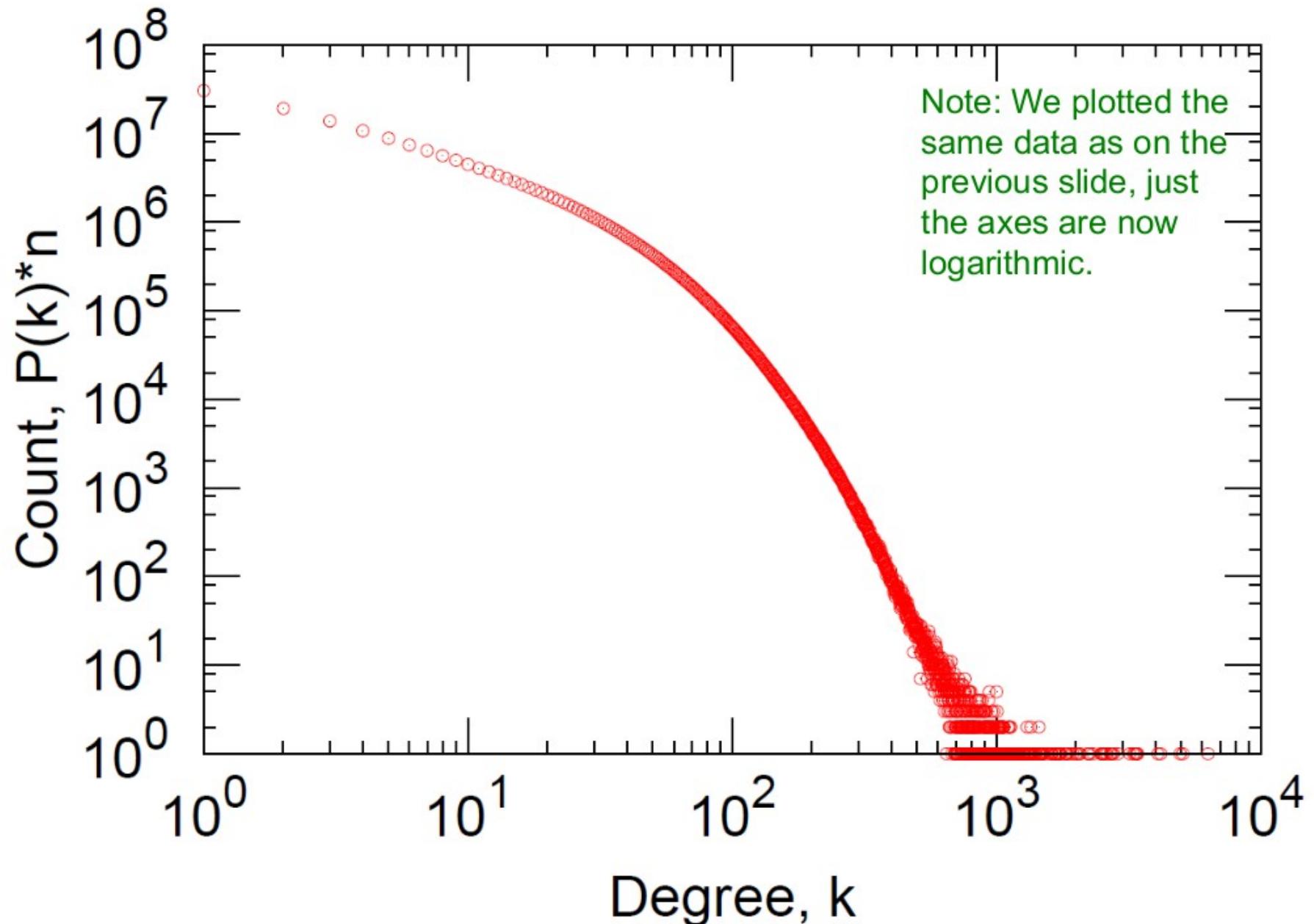
Messaging as multigraph



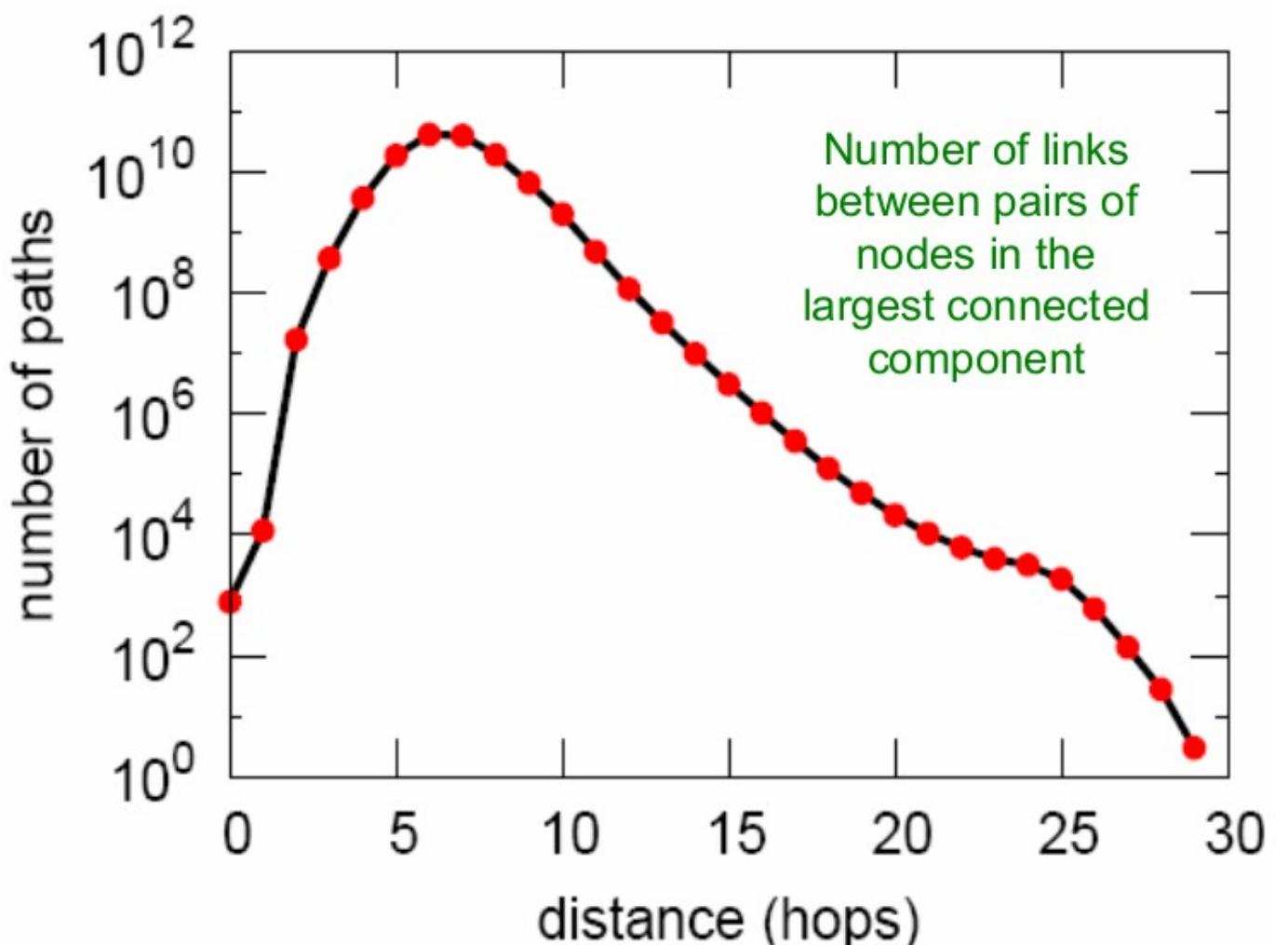
MSN: (1) Degree Distribution



MSN: Log-Log Degree Distribution



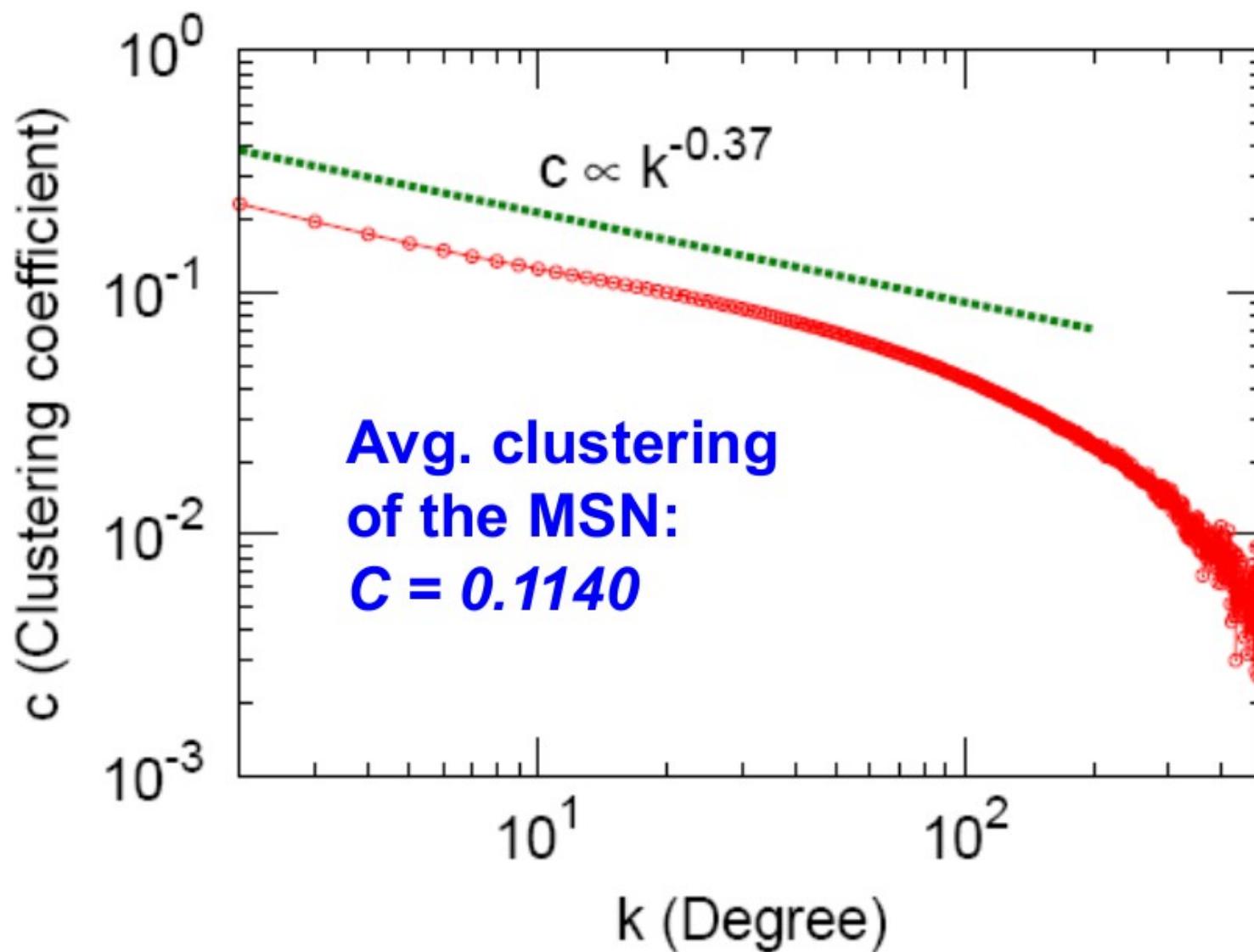
MSN: (2) Diameter



Avg. path length 6.6
90% of the nodes can be reached in < 8 hops

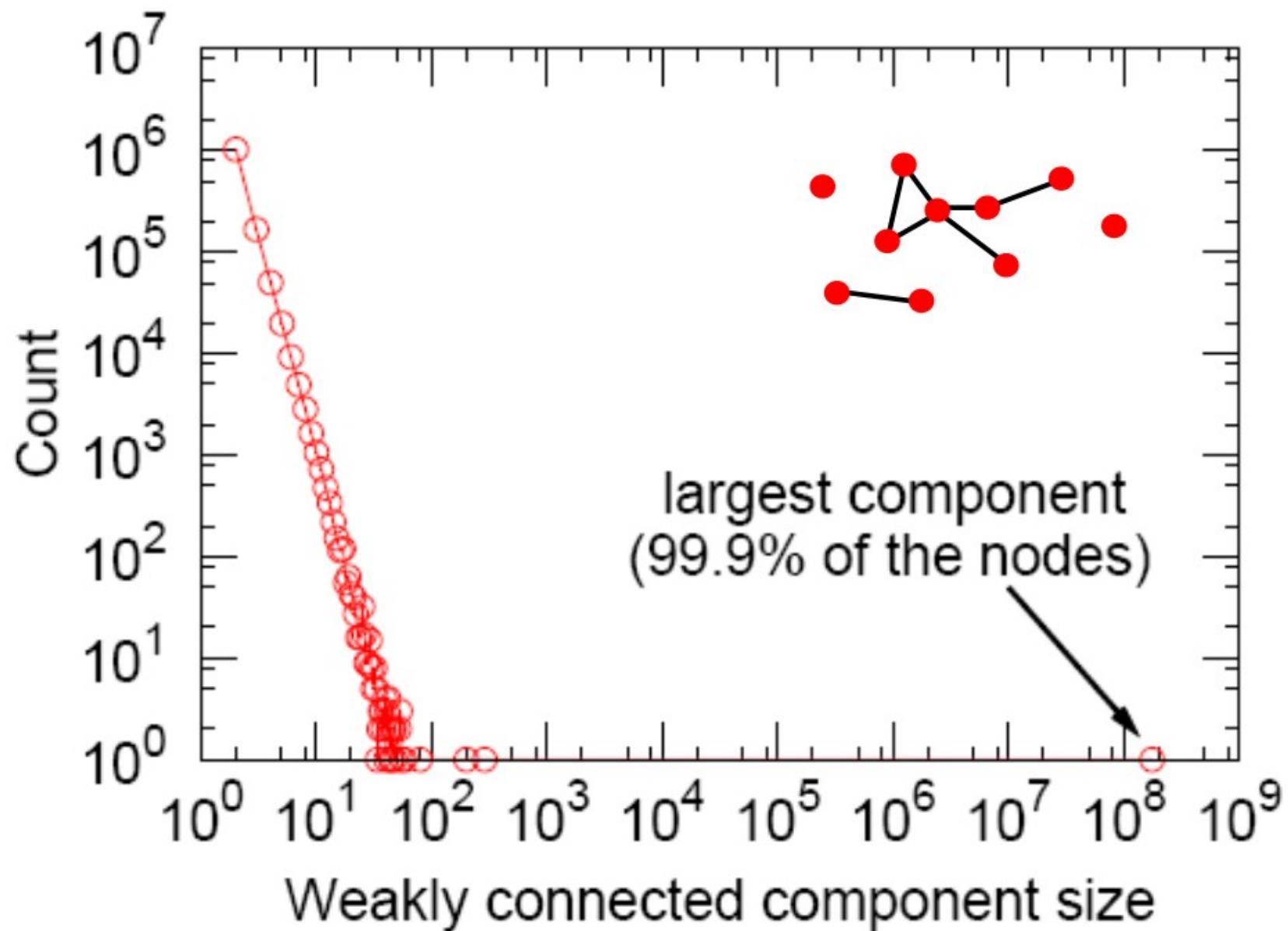
Steps	#Nodes
0	1
1	10
2	78
3	3,96
4	8,648
5	3,299,252
6	28,395,849
7	79,059,497
8	52,995,778
9	10,321,008
10	1,955,007
11	518,410
12	149,945
13	44,616
14	13,740
15	4,476
16	1,542
17	536
18	167
19	71
20	29
21	16
22	10
23	3
24	2
25	3

MSN: (3) Clustering Coefficient



$$C_k: \text{average } C_i \text{ of nodes } i \text{ of degree } k: C_k = \frac{1}{N_k} \sum_{i:k_i=k} C_i$$

MSN: (4) Connected Components



MSN: Key Network Properties

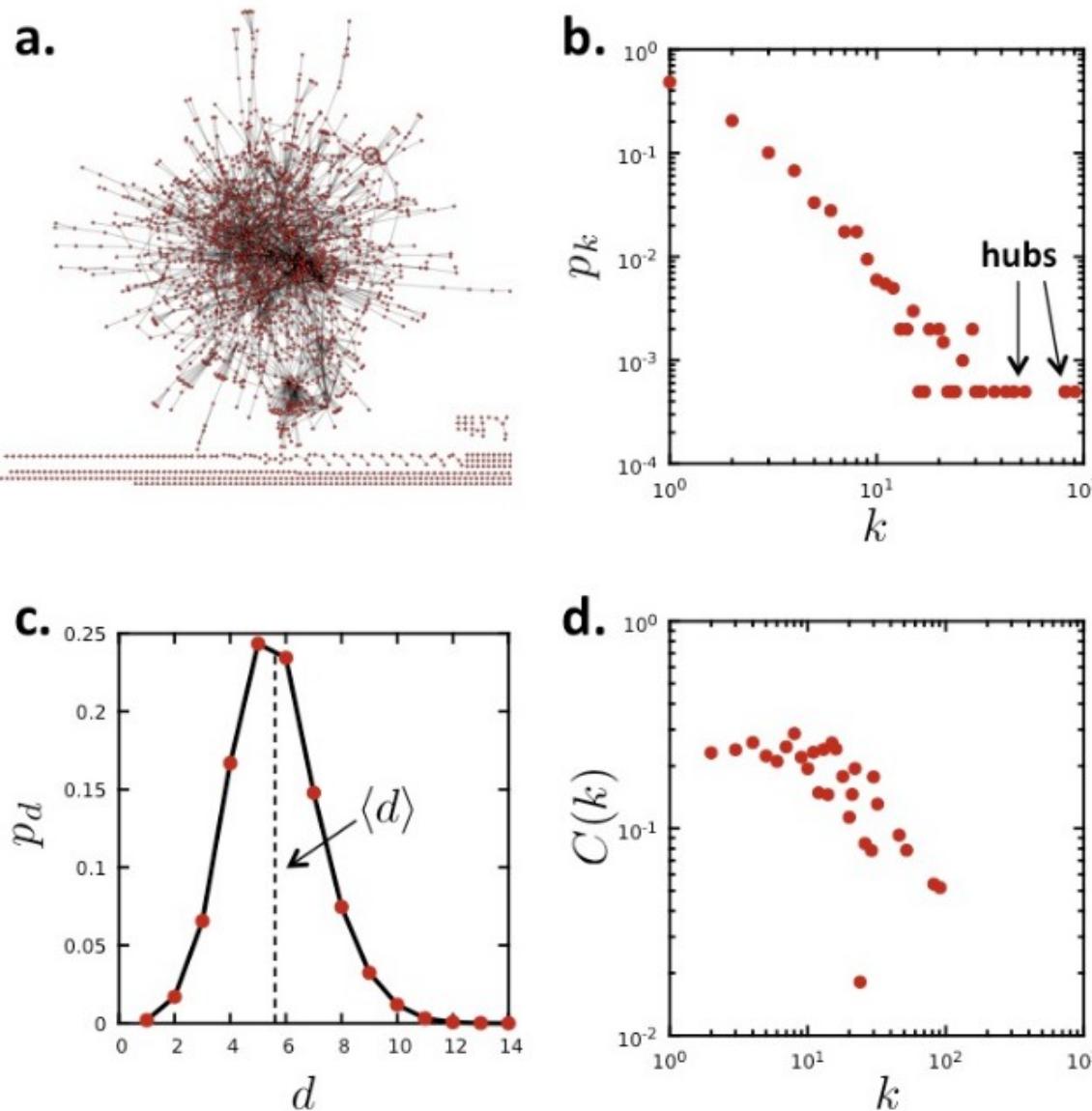
- (1) Degree distribution *Heavily skewed
avg. degree = 14.4*
- (2) Path Length **6.6**
- (3) Clustering coefficient **0.11**
- (4) Connected components *giant component*

Are these values “expected”?

Are they “surprising”?

To answer this we need a null-model!

Another Example: PPI Network



a. Undirected network

$N=2,018$ proteins as nodes

$E=2,930$ binding interactions as links.

b. Degree distribution:

Skewed. Average degree $\langle k \rangle = 2.90$

c. Diameter:

Avg. path length = 5.8

d. Clustering:

Avg. clustering = 0.12

Connectivity: 185 components
the largest component 1,647 nodes (81% of nodes)

Intermezzo: Network Datasets

The KONECT Project

Networks • Statistics • Plots • Categories • Handbook

Jérôme Kunegis
University of Namur

$n = \text{Size}$

$m = \text{Volume}$

$\bar{m} = \text{Unique edge count}$

$l = \text{Loop count}$

$s = \text{Wedge count}$

$z = \text{Claw count}$

$x = \text{Cross count}$

$t = \text{Triangle count}$

$q = \text{Square count}$

$T_4 = \text{4-Tour count}$

$d_{\max} = \text{Maximum degree}$

$d = \text{Average degree}$

$p = \text{Fill}$

$\bar{m} = \text{Average edge multiplicity}$

$N = \text{Size of LCC}$

$N_s = \text{Size of LSCC}$

$\delta = \text{Diameter}$

$\delta_{0.5} = \text{50-Percentile effective diameter}$

$\delta_{0.9} = \text{90-Percentile effective diameter}$

$\delta_M = \text{Median distance}$

$\delta_m = \text{Mean distance}$

$G = \text{Gini coefficient}$

$P = \text{Balanced inequality ratio}$

$H_{\text{er}} = \text{Relative edge distribution entropy}$

$\in \mathbb{N}$

$\in \mathbb{R}^+$

$\in [0, 1]$

$\in \mathbb{R}^+$

$\in \mathbb{N}$

$\in \mathbb{N}$

$\in \mathbb{N}$

$\in \mathbb{N}$

$\in \mathbb{R}^+$

$\in \mathbb{N}$

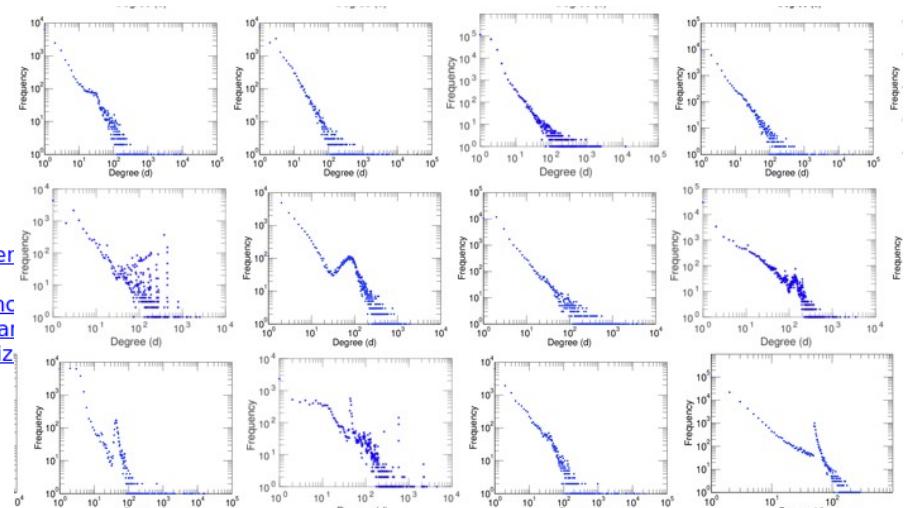
$\in \mathbb{R}^+$

$\in [0, 1]$

$\in [0, 1]$

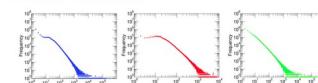
$\in [0, 1]$

- [Fruchterman-Reingold graph drawing](#)
- [Degree distribution](#)
- [Cumulative degree distribution](#)
- [Lorenz curve](#)
- [Spectral distribution of the adjacency matrix](#)
- [Spectral distribution of the normalized adjacency matrix](#)
- [Spectral distribution of the Laplacian](#)
- [Spectral graph drawing based on the adjacency matrix](#)
- [Spectral graph drawing based on the Laplacian](#)
- [Spectral graph drawing based on the normalized adjacency matrix](#)
- [Degree assortativity](#)
- [Zipf plot](#)
- [Hop distribution](#)
- [Double Laplacian graph drawing](#)
- [Delaunay graph drawing](#)
- [In/outdegree scatter plot](#)
- [Item rating evolution](#)
- [Edge weight/multiplicity distribution](#)
- [Clustering coefficient distribution](#)
- [Average neighbor degree distribution](#)
- [Temporal distribution](#)
- [Temporal hop distribution](#)
- [Diameter/density evolution](#)
- [Signed temporal distribution](#)
- [Rating class evolution](#)
- [SynGraphy](#)
- [Inter-event distribution](#)
- [Node-level inter-event distribution](#)

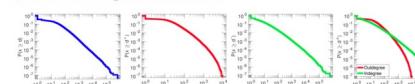


Plots

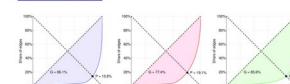
Degree distribution



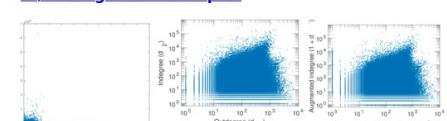
Cumulative degree distribution



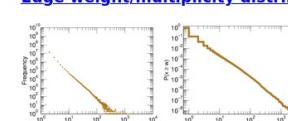
Lorenz curve



In/outdegree scatter plot



Edge weight/multiplicity distribution



<http://konect.cc/>

Intermezzo: Network Datasets

Network Repository. An *Interactive Scientific* Network Data Repository.

THE FIRST SCIENTIFIC NETWORK DATA REPOSITORY WITH INTERACTIVE VISUAL ANALYTICS.

NEW GraphVis: interactive visual graph mining and machine learning



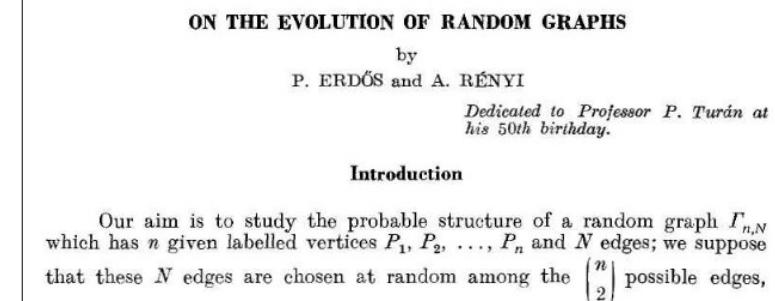
<http://networkrepository.com/>

Erdös-Renyi Random Graph Model

Simplest Model of Graphs

- Erdös-Renyi
Random Graphs

[Erdös-Renyi, '60]

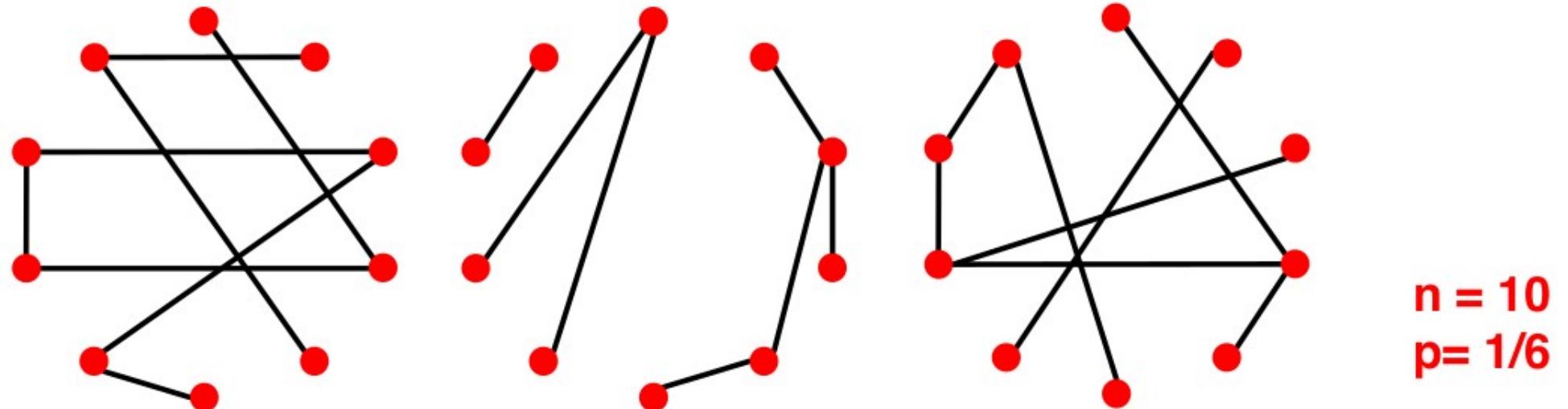


- $G_{n,p}$: undirected graph on n nodes and each (u,v) appears i.i.d. with probability p
- $G_{n,m}$: undirected graph with n nodes and m uniformly at random picked edges

What kind of networks do such models produce?

Random Graph Model

- n and p do not uniquely determine the graph!
 - The graph is a result of a random process
- We can have many different realizations given the same n and p



Properties of $G_{n,p}$

- Degree distribution $P(k)$
- Clustering coefficient C
- Path Length h
- Connected components S

What are the values of
these properties for $G_{n,p}$?

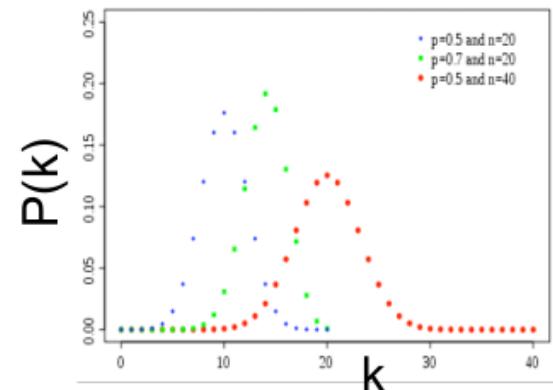
$G_{n,p}$: degree distribution

- Fact: Degree Distribution of $G_{n,p}$ is **binomial**
- Let $P(k)$ denote the fraction of nodes with degree k

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Diagram annotations:

- Select k nodes out of $n-1$ (points to the binomial coefficient term)
- Probability of having k edges (points to p^k)
- Probability of missing the rest of the $n-1-k$ edges (points to $(1-p)^{n-1-k}$)



Mean, variance of a binomial distribution

$$\bar{k} = p(n-1)$$

$$\sigma^2 = p(1-p)(n-1)$$

$$\frac{\sigma}{\bar{k}} = \left[\frac{1-p}{p} \frac{1}{(n-1)} \right]^{1/2} \approx \frac{1}{(n-1)^{1/2}}$$

By the law of large numbers, as the network size increases, the distribution becomes increasingly narrow—we are increasingly confident that the degree of a node is in the vicinity of \bar{k} .

Intermezzo: NetLogo

NetLogo

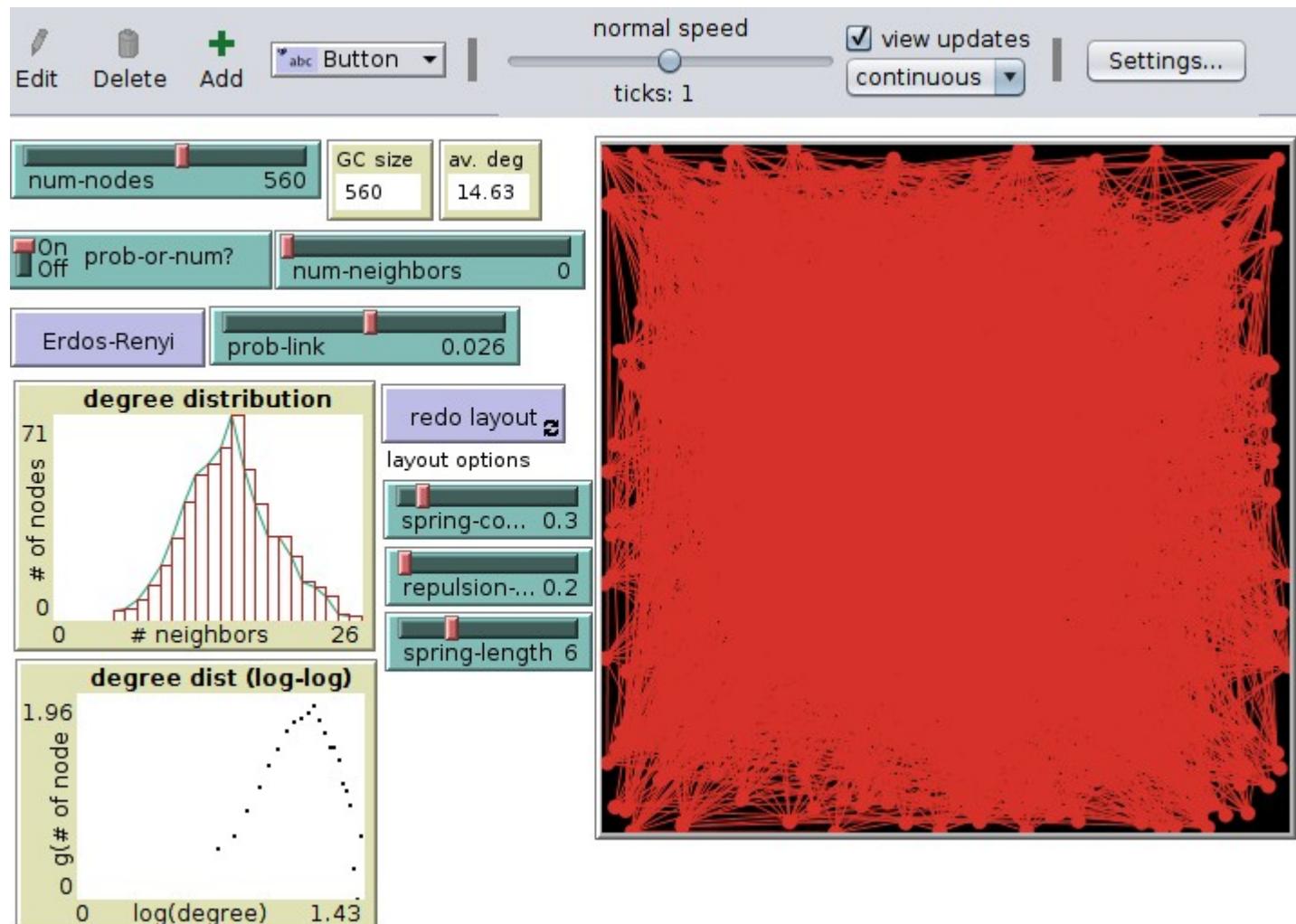
[Home](#)
[Download](#)
[Help](#)
[Resources](#)
[Extensions](#)

NetLogo is a multi-agent programmable modeling environment. It is used by many hundreds of thousands of students, teachers, and researchers worldwide. It also powers [HubNet](#) participatory simulations. It is authored by [Uri Wilensky](#) and developed at the [CCL](#). You can download it free of charge. You can also try it online through [NetLogo Web](#).

Visualize some of the properties described in the lectures

<https://ccl.northwestern.edu/netlogo/>

NetLogo: $G_{n,p}$ and degree dist.



ErdosRenyiDegDist.nlogo

$G_{n,p}$: clustering coefficient

- Remember: $C_i = \frac{2e_i}{k_i(k_i-1)}$ where e_i is the number of edges between the neighbors of node i
- Edges in $G_{n,p}$ appear i.i.d. with prob. p
- So, expected $E[e_i]$ is $= p \frac{k_i(k_i-1)}{2}$
 - each pair is connected with prob. p
 - number of distinct pairs of neighbors of node i of degree k_i
- Therefore $E[C] = \frac{p \cdot k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{n-1} \approx \frac{\bar{k}}{n}$

Clustering coefficient of a random graph is small.

If we generate bigger and bigger graphs with fixed avg. degree k (that is we set $p = k \cdot 1/n$), then C decreases with the graph size n .

Properties of $G_{n,p}$

- Degree distribution
- Clustering coefficient
- Path Length
- Connected components

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

$$C = p \approx \frac{\bar{k}}{n}$$

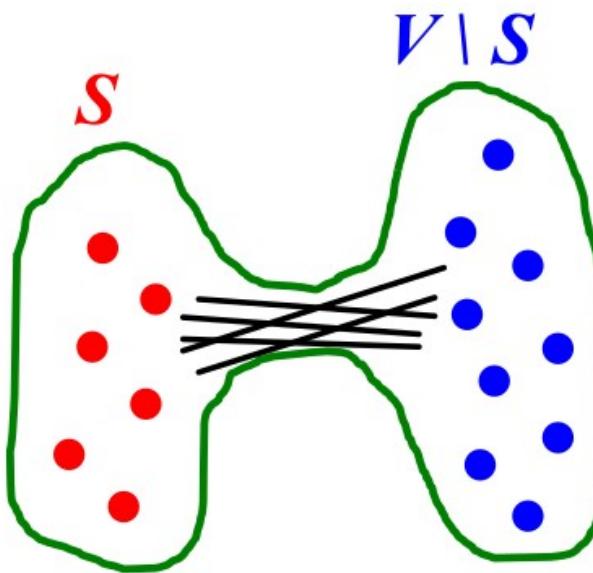
next!

What are the values of
these properties for $G_{n,p}$?

Definition: expansion

- Graph $G(V,E)$ has **expansion α** : if $\forall S \subseteq V$:
of edges leaving $S \geq \alpha \cdot \min(|S|, |V \setminus S|)$
- Or equivalently:

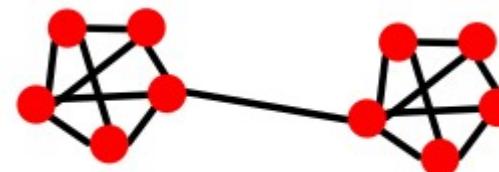
$$\alpha = \min_{S \subseteq V} \frac{\#\text{edges leaving } S}{\min(|S|, |V \setminus S|)}$$



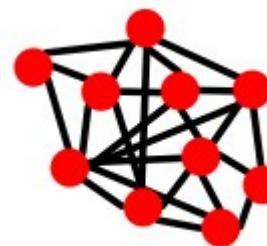
Expansion: measures robustness

- Expansion is measure of robustness:
 - to disconnect L nodes, we need to $cut \geq \alpha \cdot L \text{ edges}$

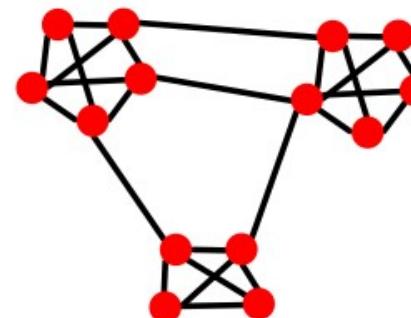
- Low expansion



- High Expansion

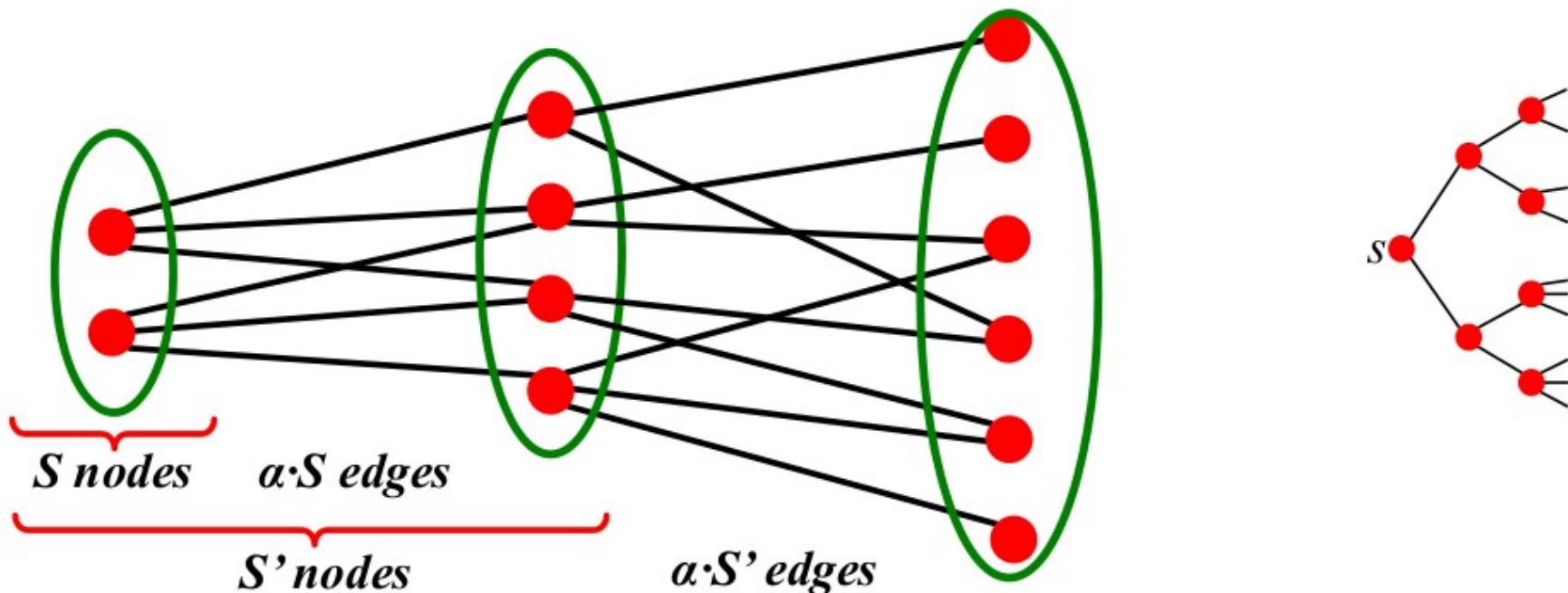


- Social Networks:
 - “communities”



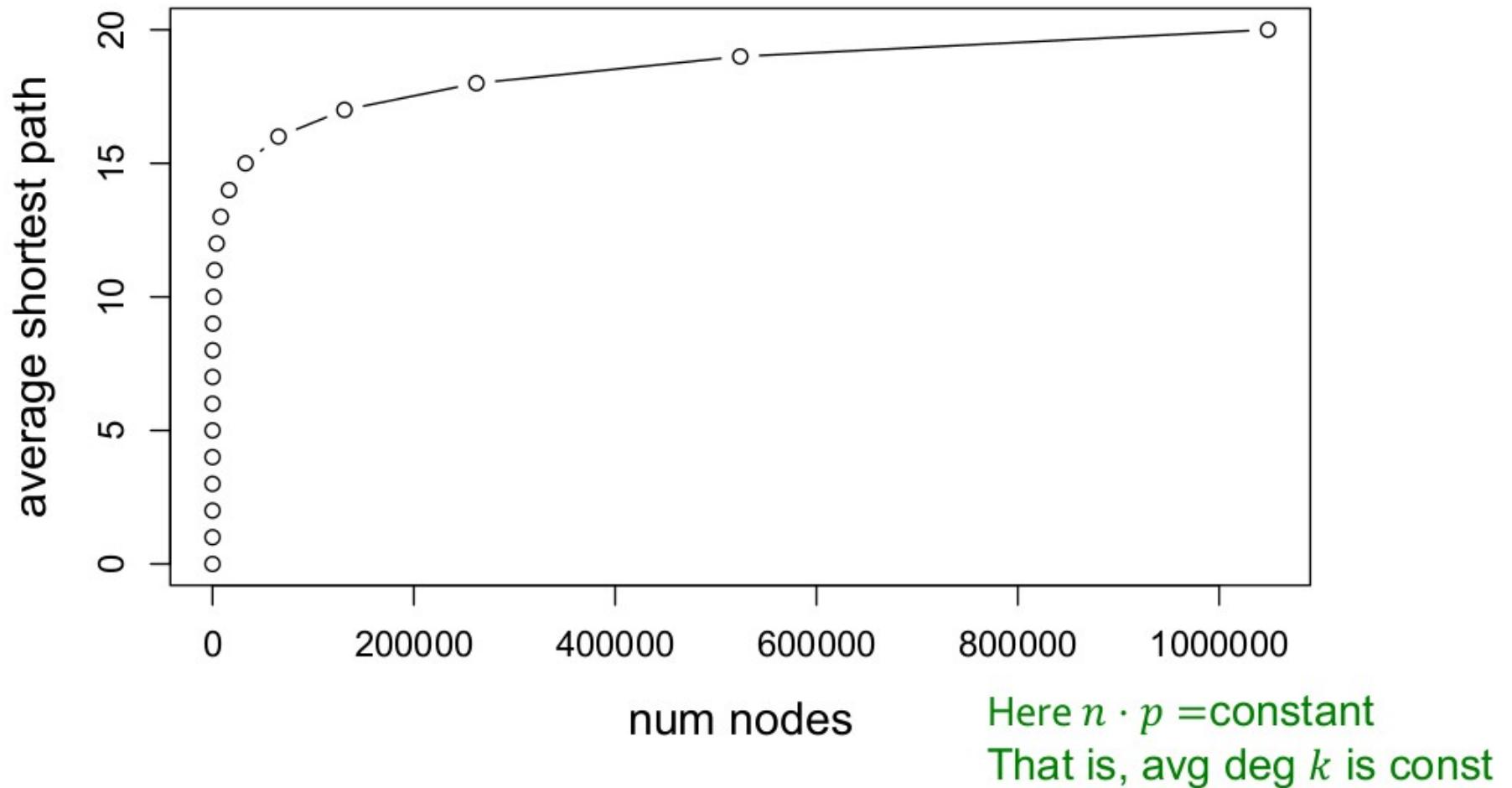
Expansion: $G_{n,p}$

- Fact: In a graph of n nodes with expansion α for all pairs of nodes there is a path of length $O((\log n)/\alpha)$.
- Random graph $G_{n,p}$:
For $\log n > np > c$, $\text{diam}(G_{n,p}) = O(\log n / \log(np))$
 - random graphs have good expansion, so it takes a logarithmic number of steps for BFS to visit all nodes



$G_{n,p}$: average shortest path

Erdös-Renyi Random Graphs can grow very large but nodes will be just a few hops apart



Properties of $G_{n,p}$

- Degree distribution

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

- Clustering coefficient

$$C = p \approx \frac{\bar{k}}{n}$$

- Path Length

$$O(\log n)$$

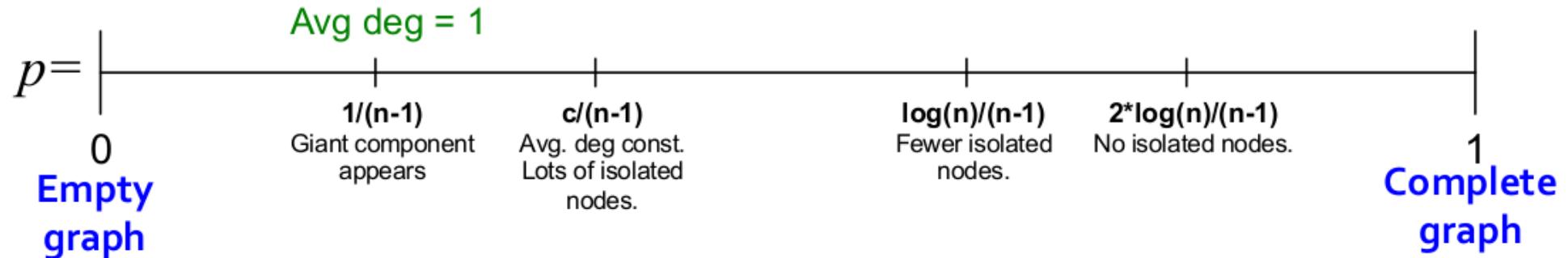
- Connected components

next!

What are the values of
these properties for $G_{n,p}$?

“Evolution” of a random graph

- Graph structure of $G_{n,p}$ as p changes

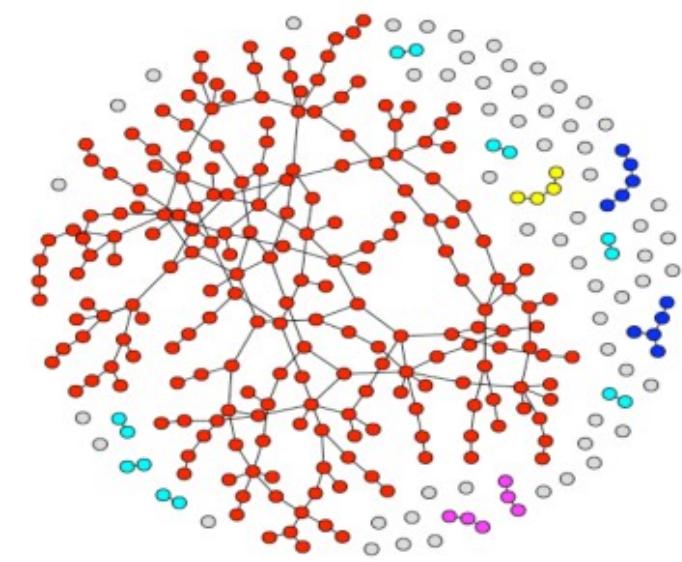
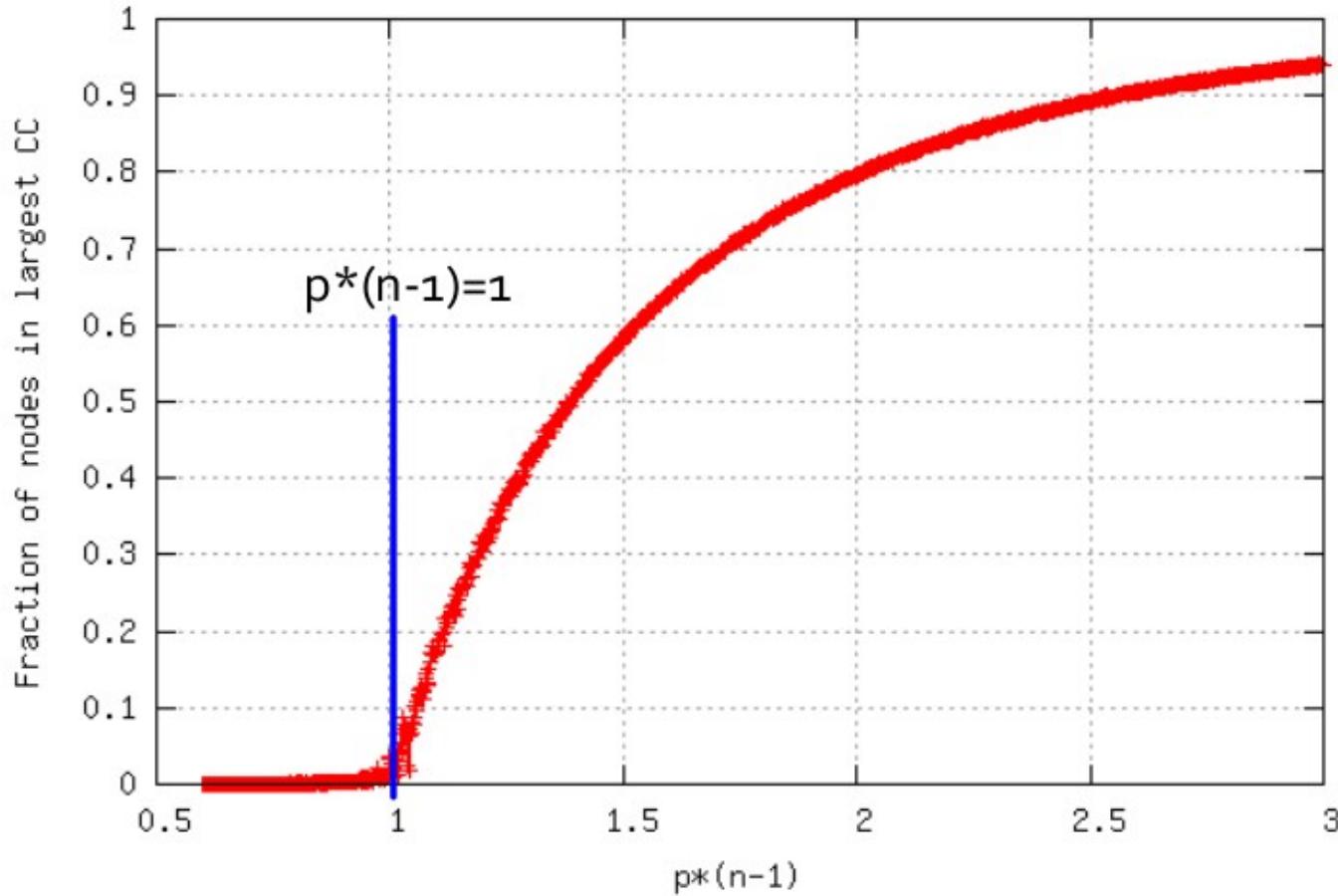


- Emergence of a **giant component**

avg. degree $k=2E/n$ or $p=k/(n-1)$

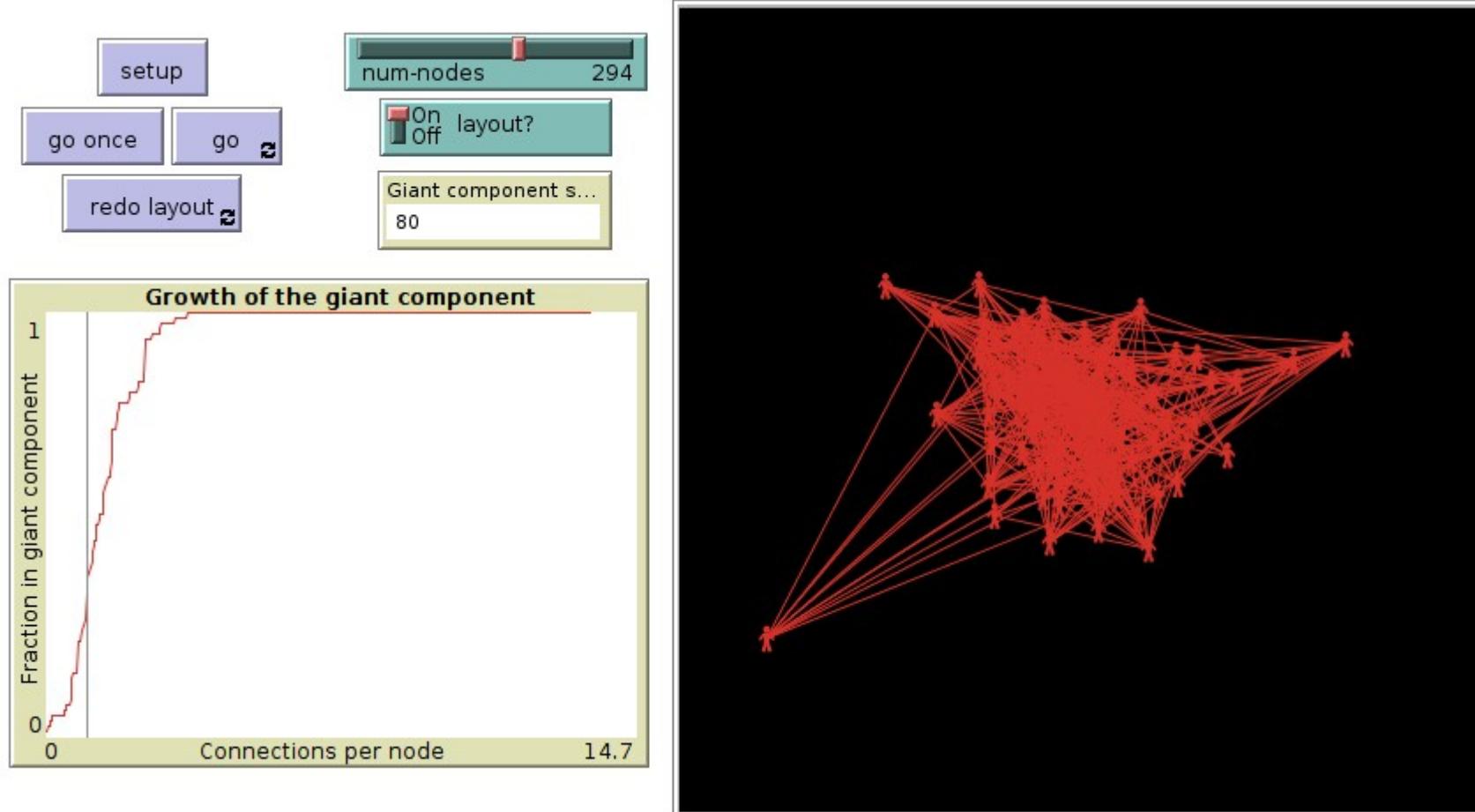
- $k=1-\varepsilon$: all components are of size $\Omega(\log n)$
- $k=1+\varepsilon$: 1 component of size $\Omega(n)$, others have size $\Omega(\log n)$
 - Each node has at least one edge in expectation

$G_{n,p}$ Simulation Experiment



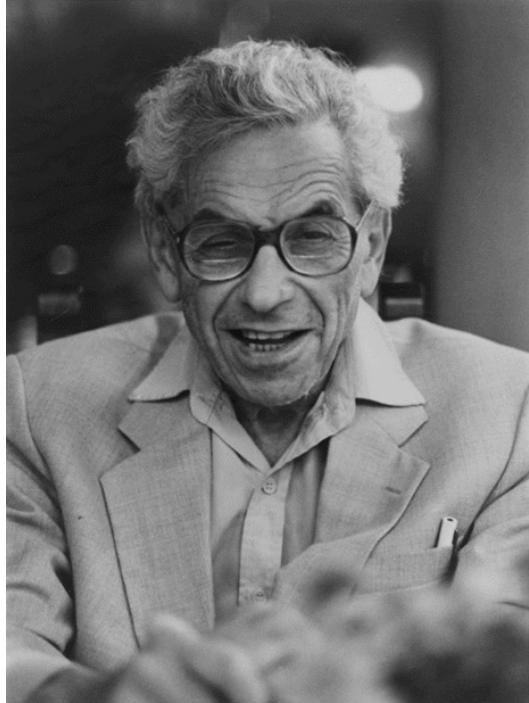
- $G_{n,p}, n=10^6, k=p(n-1) = 0.5 \dots 3$

NetLogo: $G_{n,p}$ and giant component



GiantComponent.nlogo

$G_{n,p}$ - Erdös-Renyi Model



"[When asked why are numbers beautiful?]

It's like asking why is Ludwig van Beethoven's Ninth Symphony beautiful. If you don't see why, someone can't tell you. I know numbers are beautiful. If they aren't beautiful, nothing is."

— Paul Erdos

Paul Erdős, the most prolific mathematician who ever lived, has no home and no job, but he has wandered the world for over fifty years, inspiring other mathematicians. From the documentary *N is a Number: A Portrait of Paul Erdős* © 1993 by George Csicsery

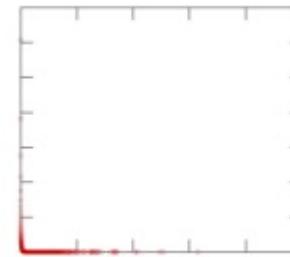
- $G_{n,p}$ is a cool model!

But let's compare it to real world networks

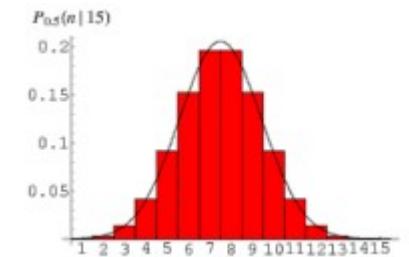
$MSN \text{ vs } G_{n,p}$

- Degree distribution

MSN



$G_{n,p}$



- Avg. Clustering coef.

0.11

\bar{k}/n
 $C \approx 8 \cdot 10^{-8}$



- Path Length

6.6

$O(\log n)$



- Largest Conn. Comp.

99%

GCC exists
when $\bar{k} > 1$
 $\bar{k} \approx 14$



Real Networks vs $G_{n,p}$

- Are real networks like random graphs?
 - Average Path Length
 - Giant Connected Component
 - Degree Distribution
 - Clustering Coefficient
- **Problems** with the random networks model:
 - Degree distribution differs from that of real networks
 - Clustering Coefficient is much lower than on real networks
 - Giant component in most real networks does NOT emerge through a phase transition
- Most important: **Are real networks random?**
 - The answer is simply: **NO!**

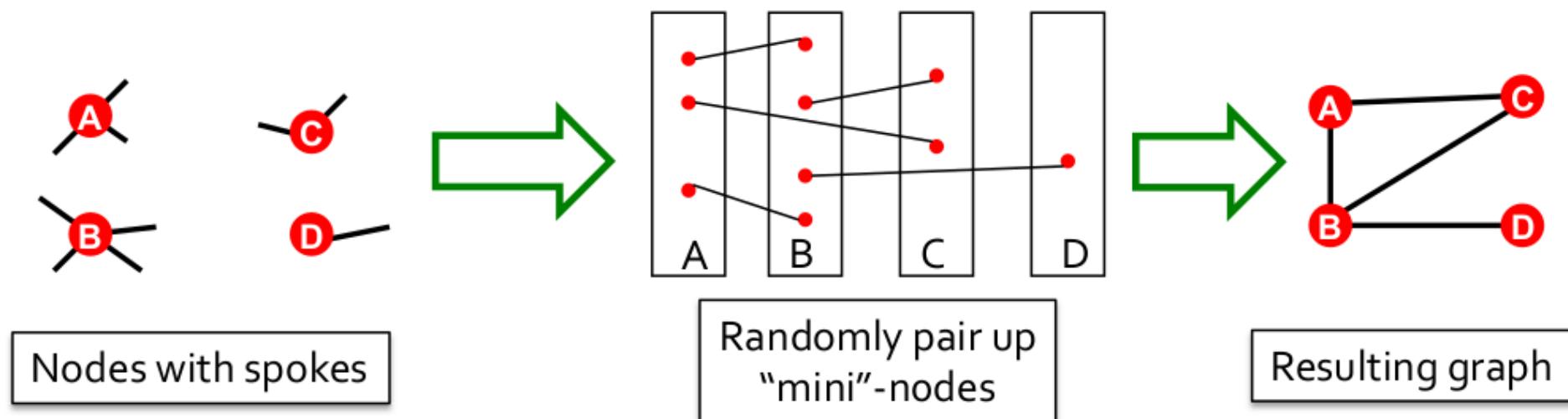
Real Networks vs $G_{n,p}$

- If $G_{n,p}$ is wrong, why did we spend time on it?
 - It is the reference model
 - It will help us calculate many quantities, that can then be compared to the real data
 - It will help us understand to what degree is a particular property the result of some random process

So, while $G_{n,p}$ is “WRONG”, it will turn out to be extremely USEFUL!

Intermezzo: Configuration Model

- Goal: Generate a random graph with a given degree sequence $k_1, k_2, \dots k_N$
- Configuration Model:



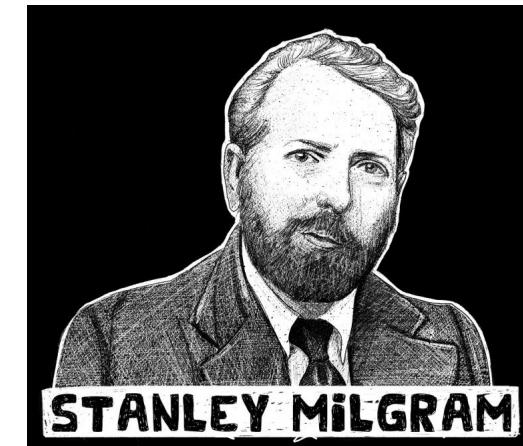
- Useful as a “null” model of networks:
 - We can compare the real network **G** and a “random” **G'** which has the same degree sequence as **G**

The Small World Random Graph Model

Can we have high clustering while also having short paths?

The Small World Experiment

- What is the **typical shortest path length** between any two persons?
 - Experiment on the global friendship network
 - Can't measure, need to probe explicitly
- **Small-world experiment**
[Milgram'67] [Travers and Milgram '69]
 - Picked 296 people in Omaha, Nebraska and Wichita, Kansas
 - Ask them to get a letter to a stock-broker in Boston by passing it through friends
- How many steps did it take?



The Small-World Problem

By Stanley Milgram

An Experimental Study of the
Small World Problem*

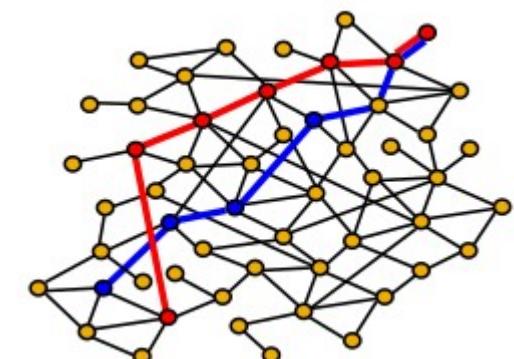
JEFFREY TRAVERS

Harvard University

AND

STANLEY MILGRAM

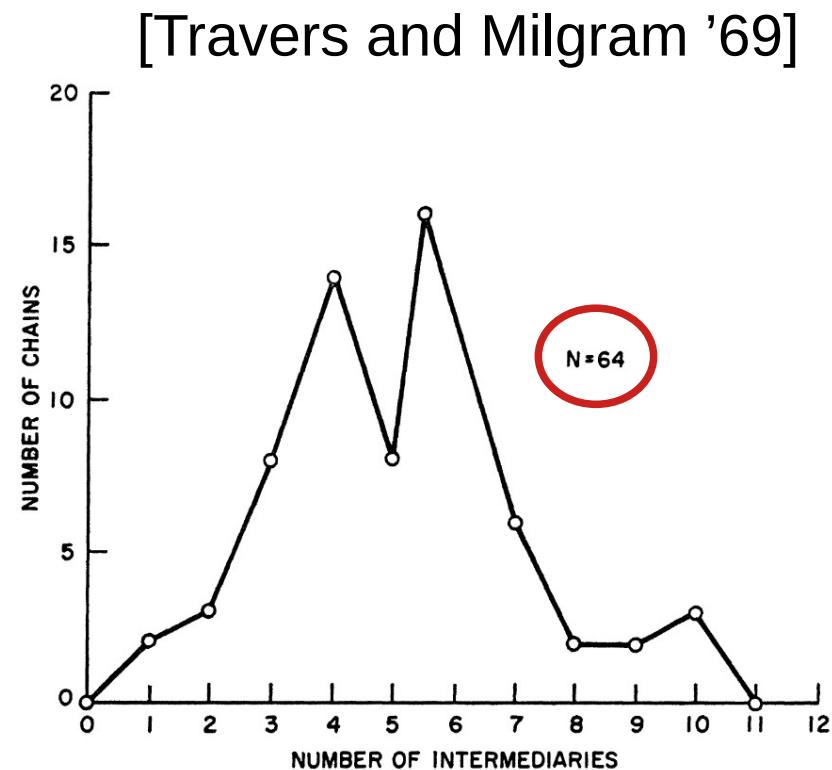
The City University of New York



The Small World Experiment

- **64 chains completed:**
(i.e., 64 letters reached the target)
 - It took 6.2 steps on the average, thus
“6 degrees of separation”

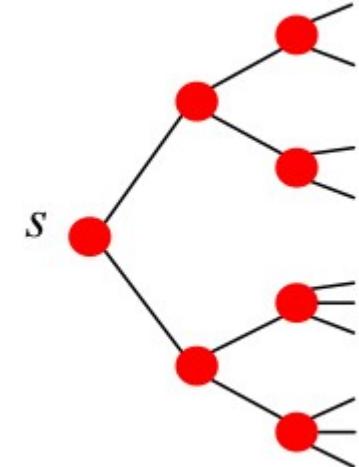
- **Further observations:**
 - People who owned stock had shorter paths to the stockbroker than random people: 5.4 vs. 6.7
 - People from the Boston area have even closer paths: 4.4



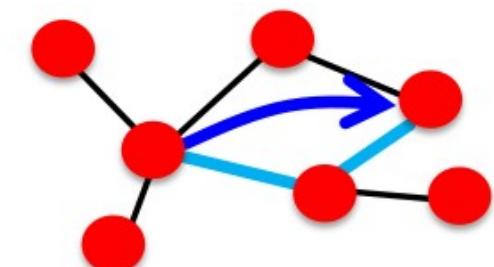
6 degrees: Should we be surprised?

- Assume each human is connected to 100 other people
Then:

- Step 1: reach 100 people
 - Step 2: reach $100 \times 100 = 10,000$ people
 - Step 3: reach $100 \times 100 \times 100 = 1M$ people
 - Step 4: reach $100 \times 100 \times 100 \times 100 = 100M$ people
 - **In 5 steps we can reach 10 billion people!**

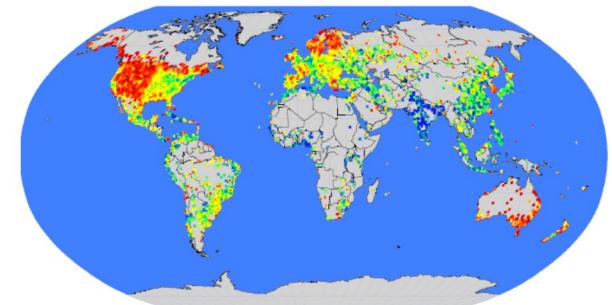


- What's wrong here? We ignore clustering!
 - Not all edges point to new people
 - 92% of FB friendships happen through a **friend-of-a-friend**



Clustering Implies Edge Locality

- MSN network has 7 orders of magnitude larger clustering than the corresponding $G_{n,p}$!



- Other Examples:

- Actor Collaborations (IMDB): $N = 225,226$ nodes, avg. degree $\bar{k} = 61$
- Electrical power grid: $N = 4,941$ nodes, $\bar{k} = 2.67$
- Network of neurons: $N = 282$ nodes, $\bar{k} = 14$

Network	h_{actual}	h_{random}	C_{actual}	C_{random}
Film actors	3.65	2.99	0.79	0.00027
Power Grid	18.70	12.40	0.080	0.005
C. elegans	2.65	2.25	0.28	0.05

h ... Average shortest path length

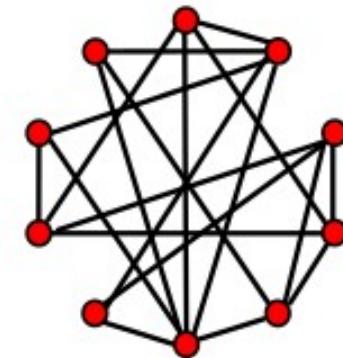
C ... Average clustering coefficient

“actual” ... real network

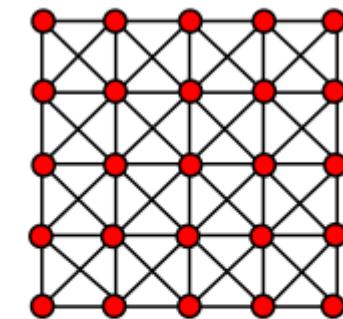
“random” ... random graph with same avg. degree

The “Controversy”

- Consequence of expansion:
 - **Short paths:** $O(\log n)$
 - This is the smallest diameter we can get if we have a constant degree.
 - But clustering is low!
- However, **networks have “local” structure:**
 - **Triadic closure:**
 - Friend of a friend is my friend
 - High clustering but diameter is also high
- **How can we have both?**



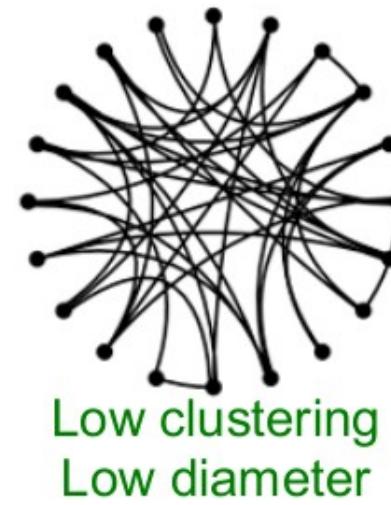
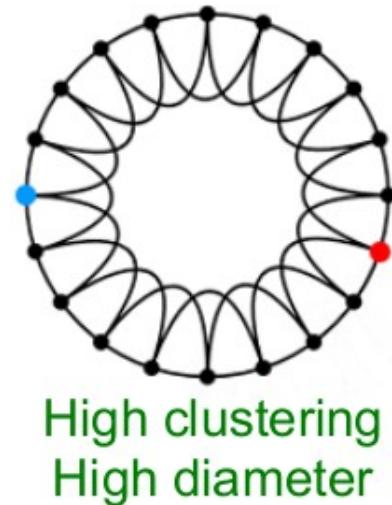
Low diameter
Low clustering coefficient



High clustering coefficient
High diameter

Small-World: How?

- Could a network with high clustering also be “small world” ($\log n$ diameter)?
 - How can we at the same time have **high clustering** and **small diameter**?



- Clustering implies edge “locality”
- Randomness enables “shortcuts”

Solution: The Small-World Model

Small-World Model

[Watts-Strogatz '98]

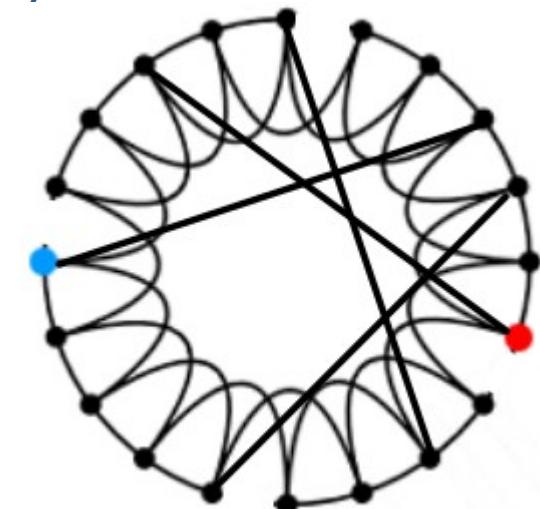
Two components to the model:

- (1) Start with a **low-dimensional regular lattice**
 - (In our case we are using a ring as a lattice)
 - Has high clustering coefficient
- Now introduce **randomness** (“shortcuts”)
- (2) **Rewire**:
 - Add/remove edges to create shortcuts to join remote parts of the lattice
 - For each edge with prob. p move the other end to a random node

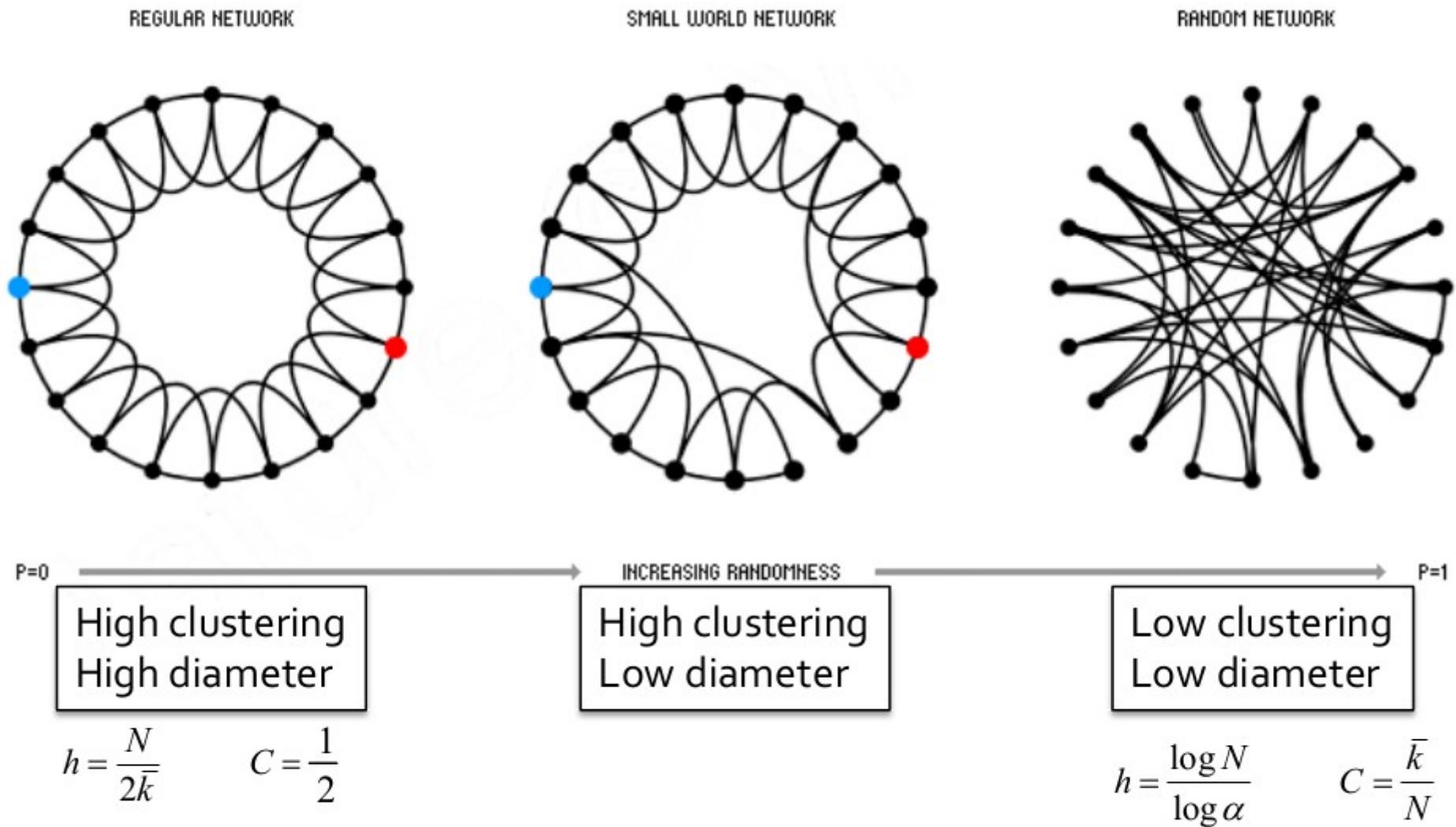
Collective dynamics of ‘small-world’ networks

Duncan J. Watts* & Steven H. Strogatz

*Department of Theoretical and Applied Mechanics, Kimball Hall,
Cornell University, Ithaca, New York 14853, USA*

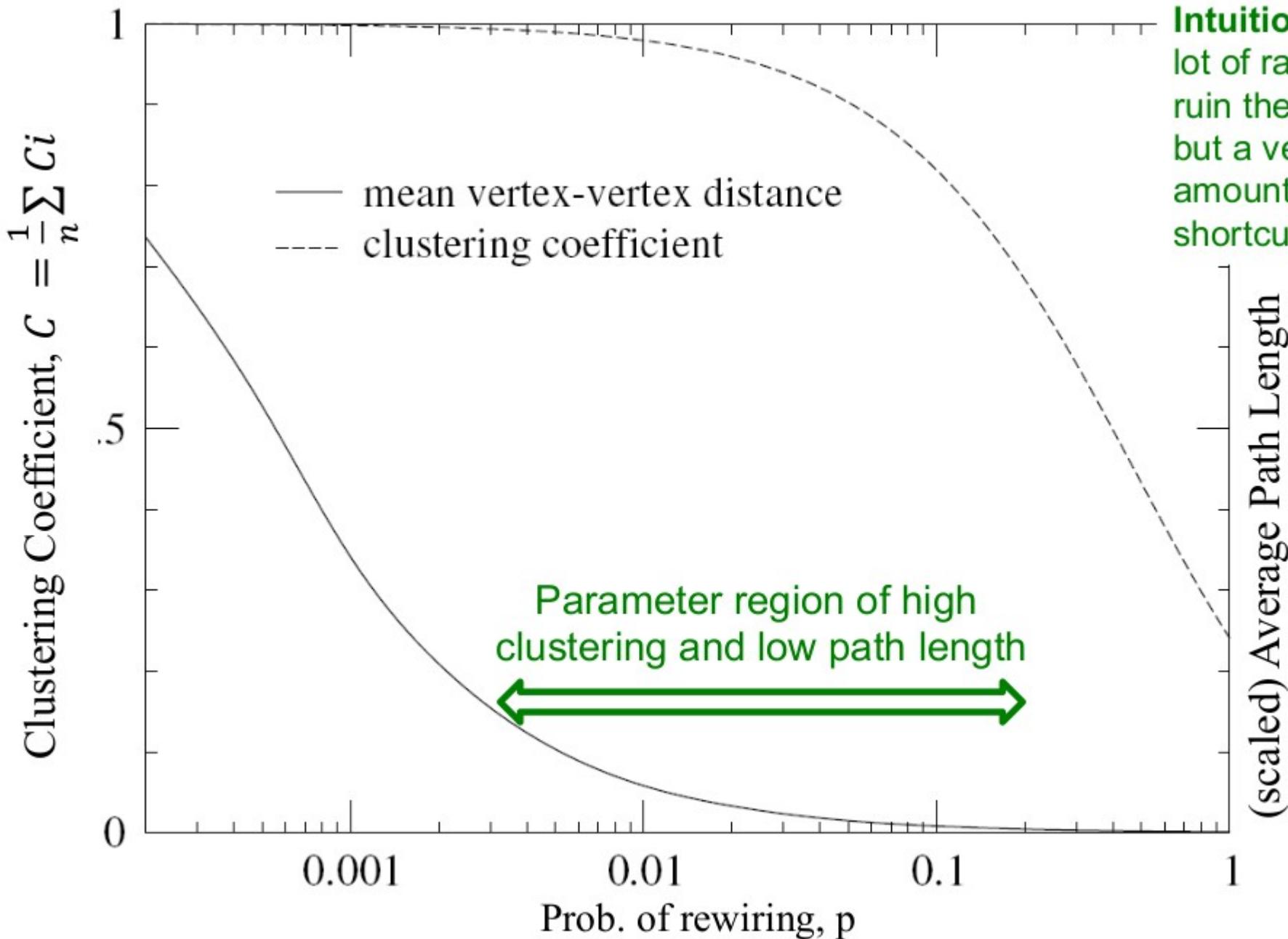


The Small World Model



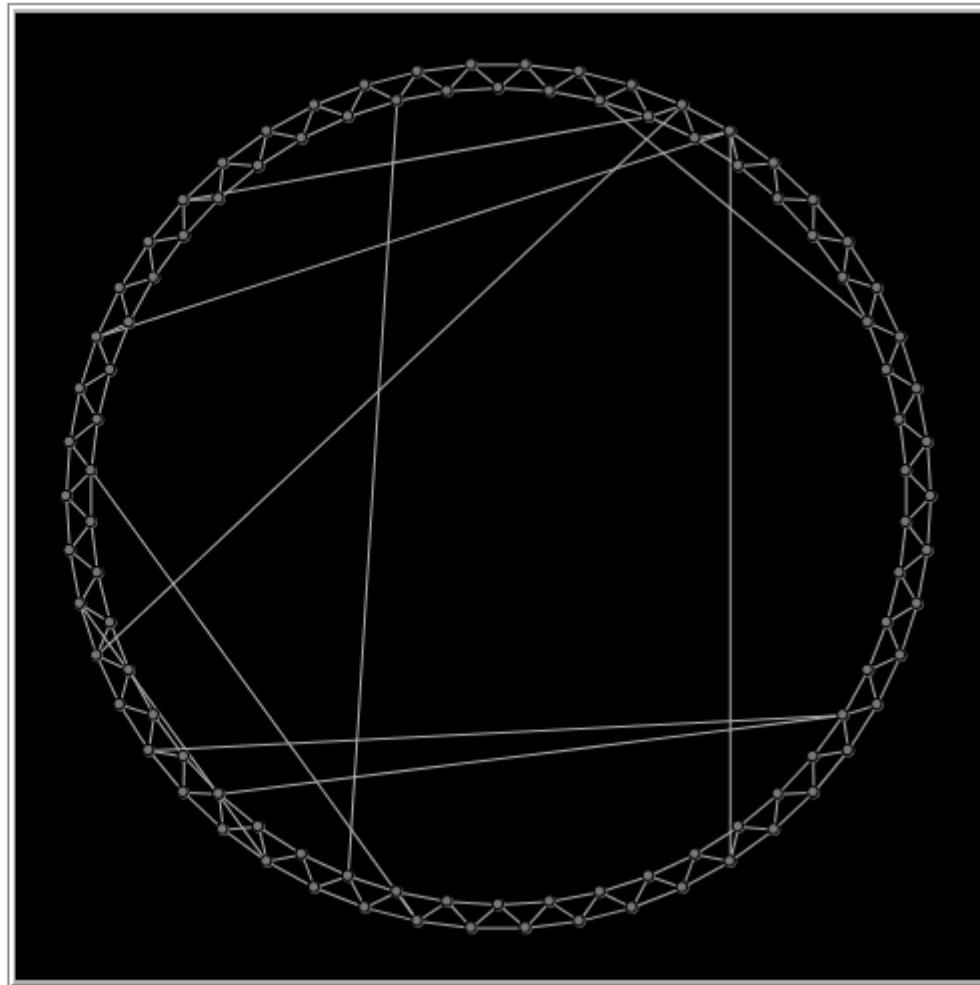
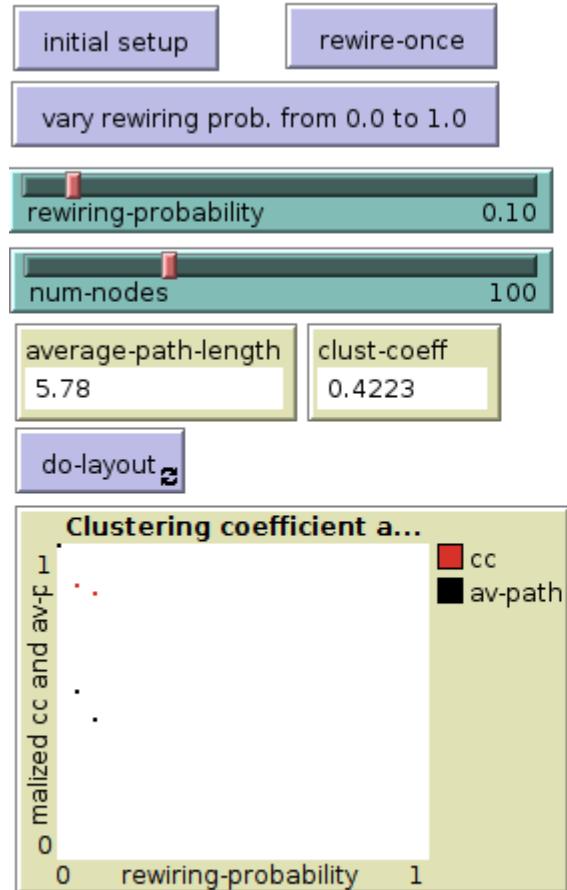
Rewiring allows us to “interpolate” between a regular lattice and a random graph

The Small World Model



Intuition: It takes a lot of randomness to ruin the clustering, but a very small amount to create shortcuts.

NetLogo: $G_{n,p}$ and Small-World



SmallWorldWS.nlogo

Small-World: Summary

- Could a network with high clustering be at the same time a “small world”?
 - Yes! You don’t need more than a few random links
- The Watts-Strogatz Model:
 - Provides insight on the interplay between clustering and being “small-world”
 - Captures the structure of many realistic networks
 - Accounts for the high clustering of real networks 
 - Does not lead to the correct degree distribution 

We usually call **small world** to networks which exhibit:

- Short avg. path length ($\log n$)
- *High clustering coefficient*

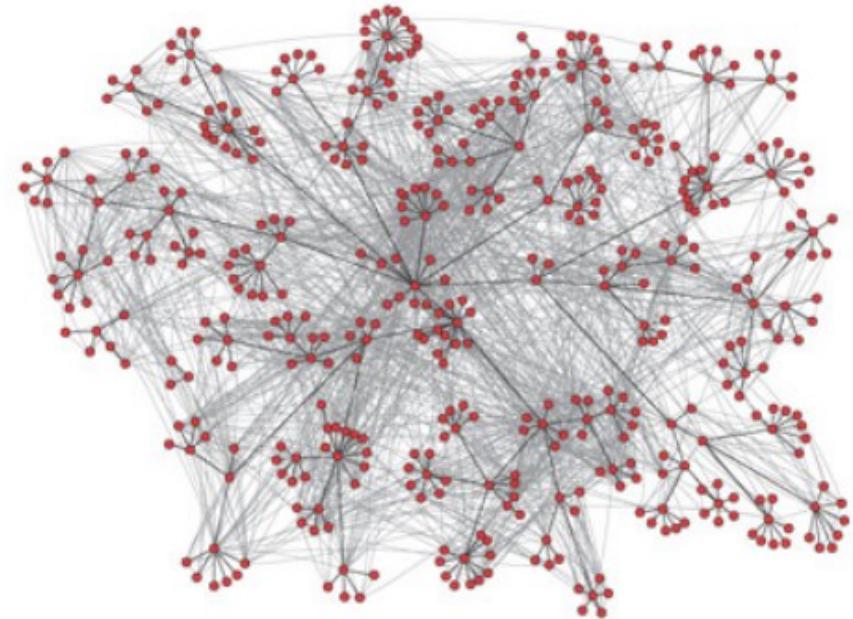
Power Laws and Degree Distributions

Realistic Degree Distribution

Which interesting graph properties do we observe that need explaining?

- Small-world model:

- Avg. Path Length
- Clustering coefficient

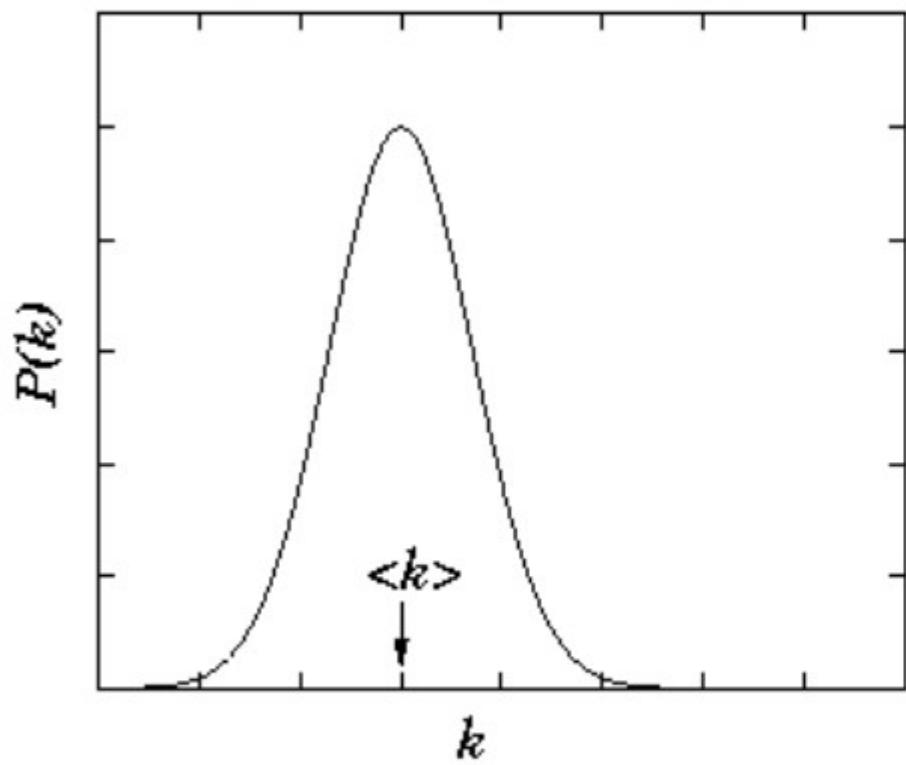


- What about **node degree distribution**?

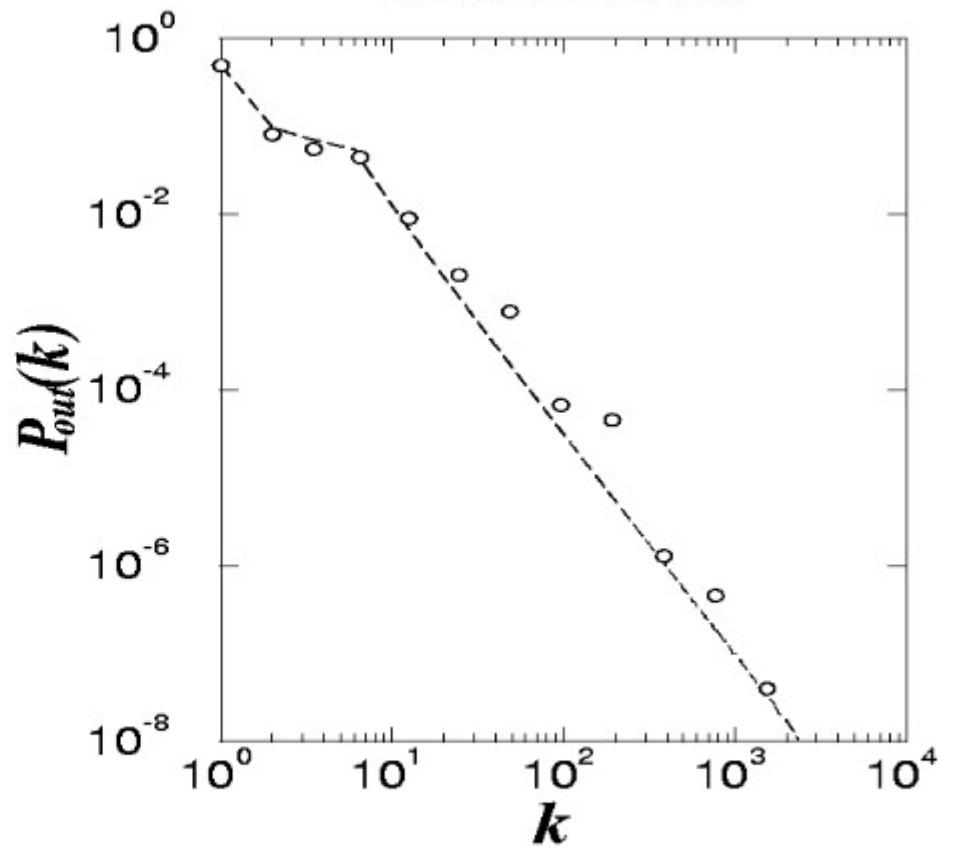
- What fraction of nodes has degree k (as a function of k)?
- Observation in **real networks**: very often a **power law**: $P(k) \propto k^{-\alpha}$
- Small-World is similar to $G_{n,p}$: **pronounced peak at k** does not result in realistic distributions...

Realistic Degree Distribution

Expected based on G_{np}

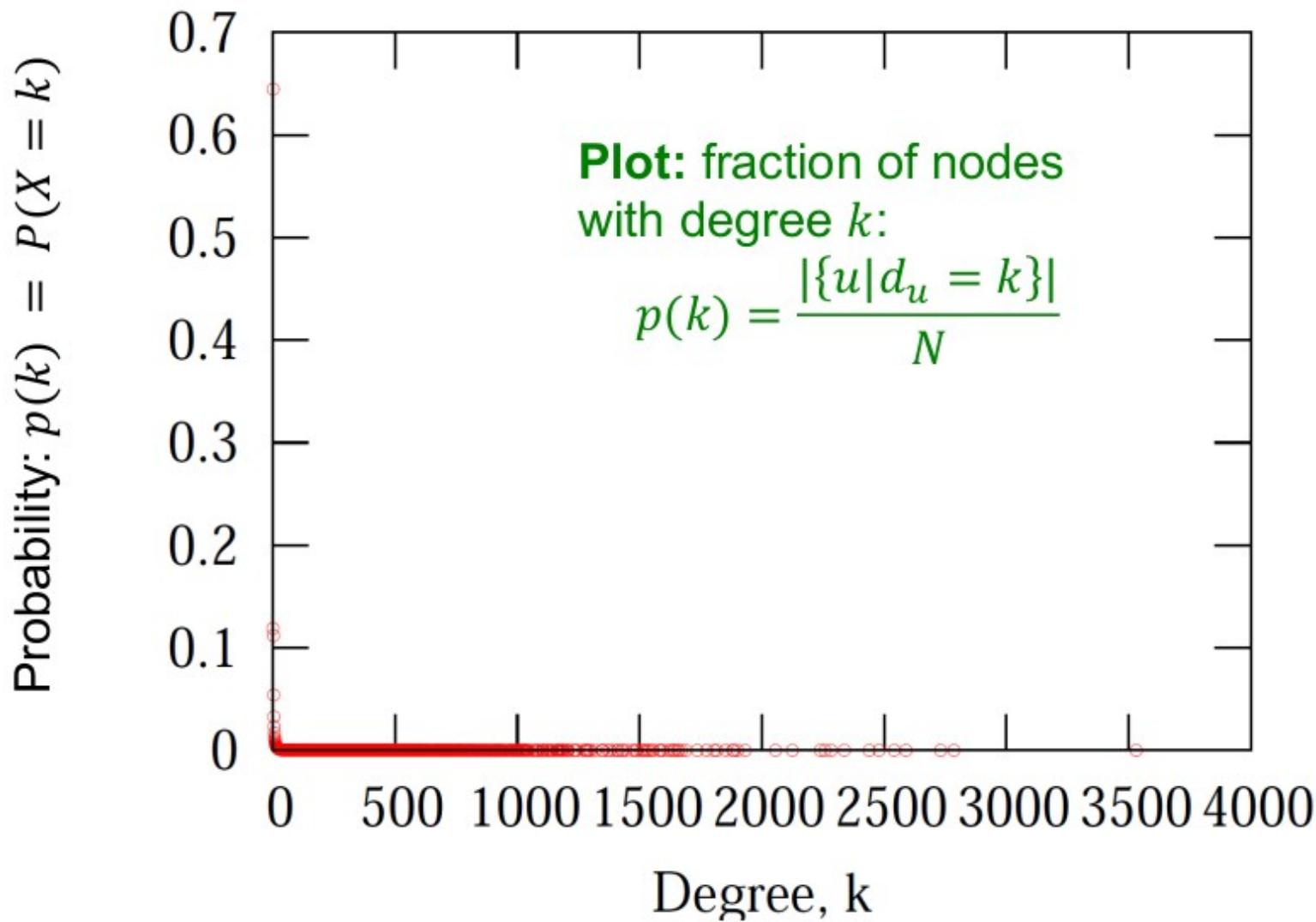


Found in data



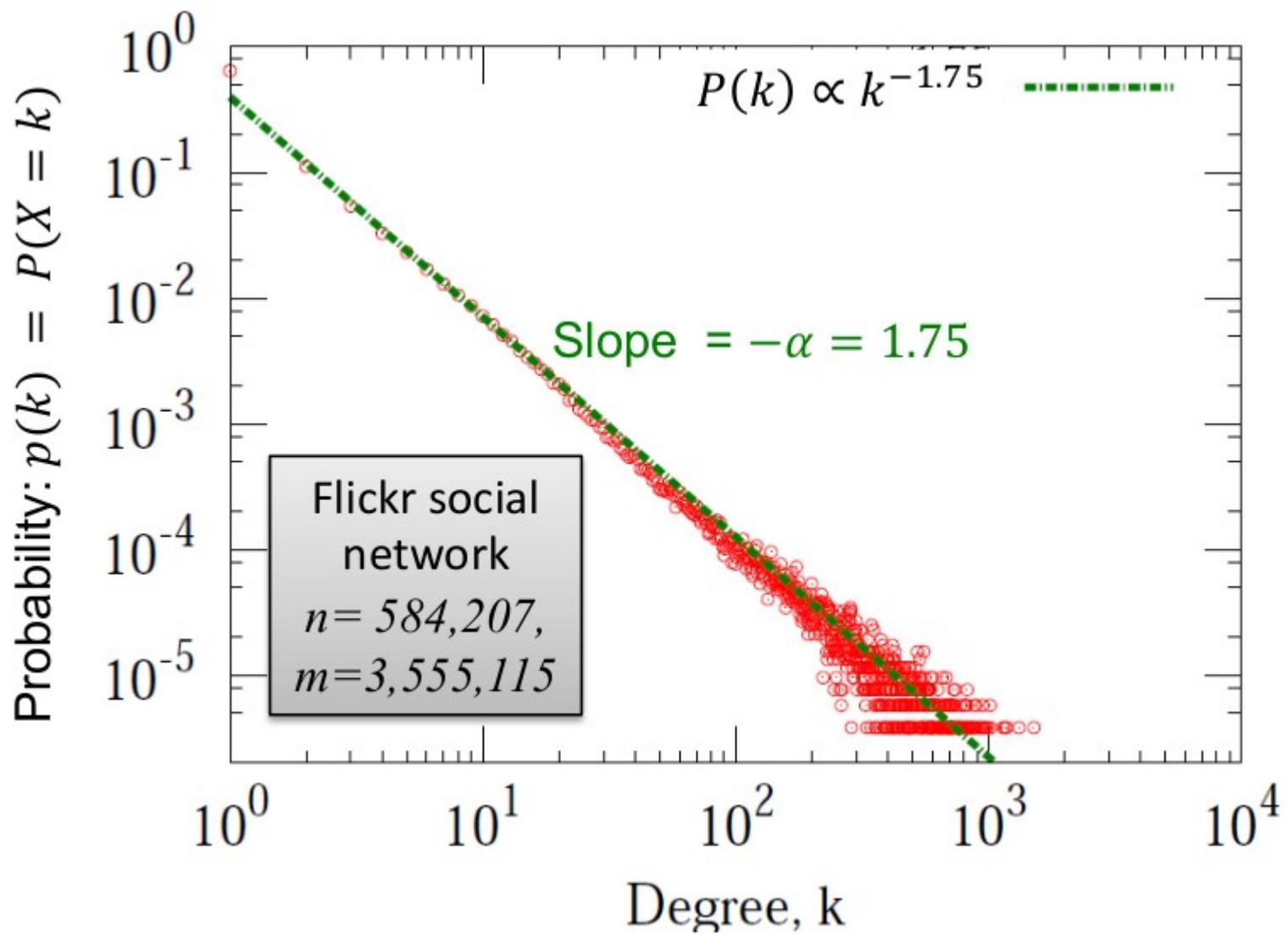
$$P(k) \propto k^{-\alpha}$$

Example: Flickr



[Leskovec et al. KDD '08]

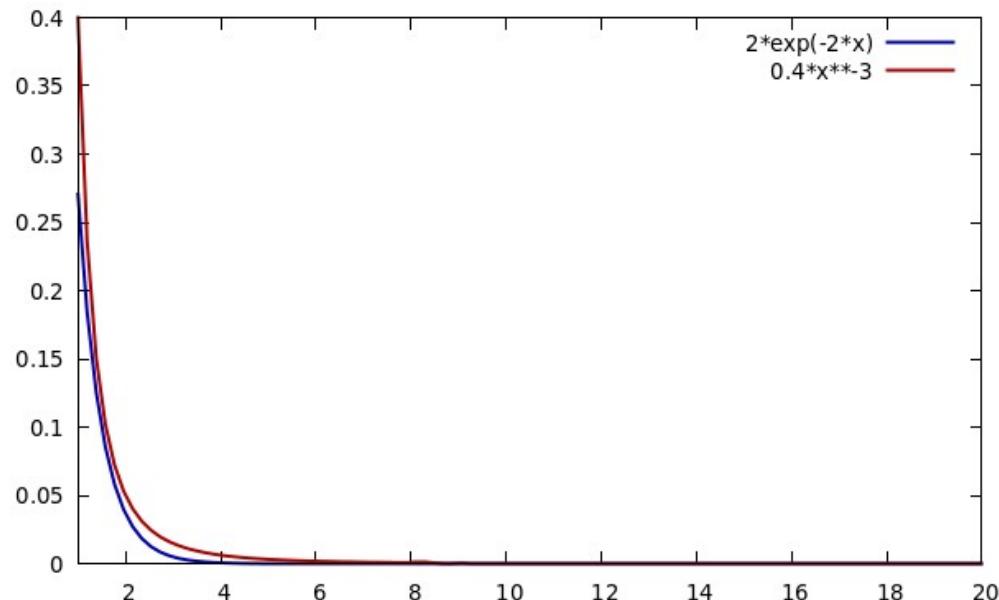
Example: Flickr



Same plot, but now on **log-log** scale

Intermezzo: exponential vs power-law

- How to distinguish:
 - **Exponential:** $P(k) \propto \lambda e^{-\lambda k}$
 - **Power-Law:** $P(k) \propto k^{-\alpha}$

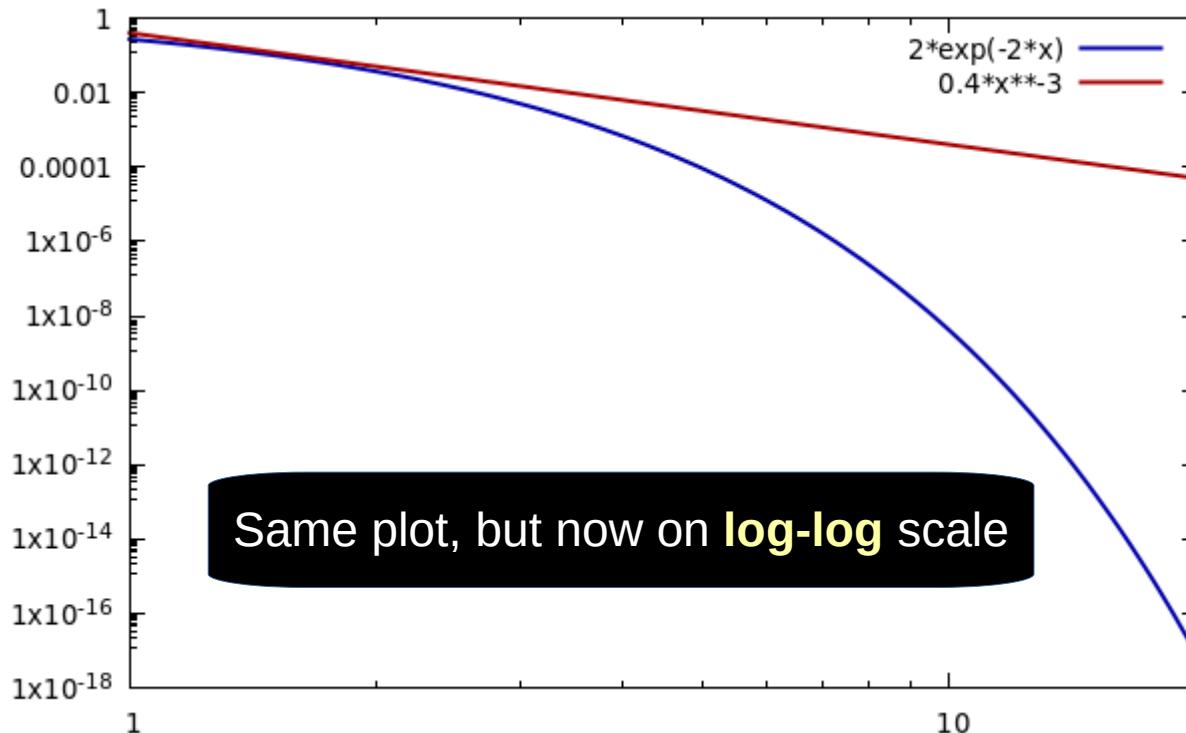


gnuplot

```
plot [1:20] 2*exp(-2*x) lt rgb "#0000aa" lw 2, 0.4*x**-3 lt rgb "#aa0000" lw 2
```

Intermezzo: exponential vs power-law

- **Exponential:** $P(k) \propto \lambda e^{-\lambda k}$
vs
- **Power-Law:** $P(k) \propto k^{-\alpha}$



If $y = f(x) = x^{-\alpha}$, then
 $\log(y) = -\alpha \log(x)$

On a log-log axis
a power law
looks like
a **straight line**
of slope $-\alpha$

gnuplot

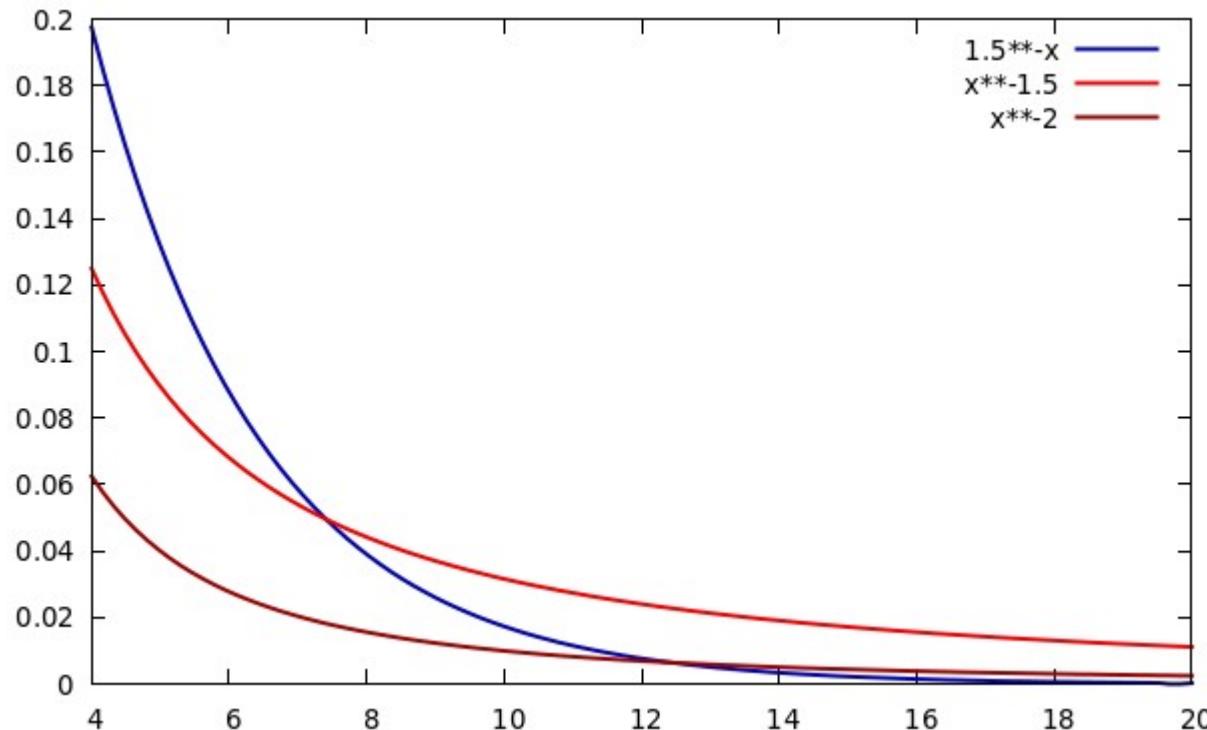
set logscale xy

Intermezzo: exponential vs power-law

- **Exponential:** $P(k) \propto \lambda e^{-\lambda k}$

vs

- **Power-Law:** $P(k) \propto k^{-\alpha}$



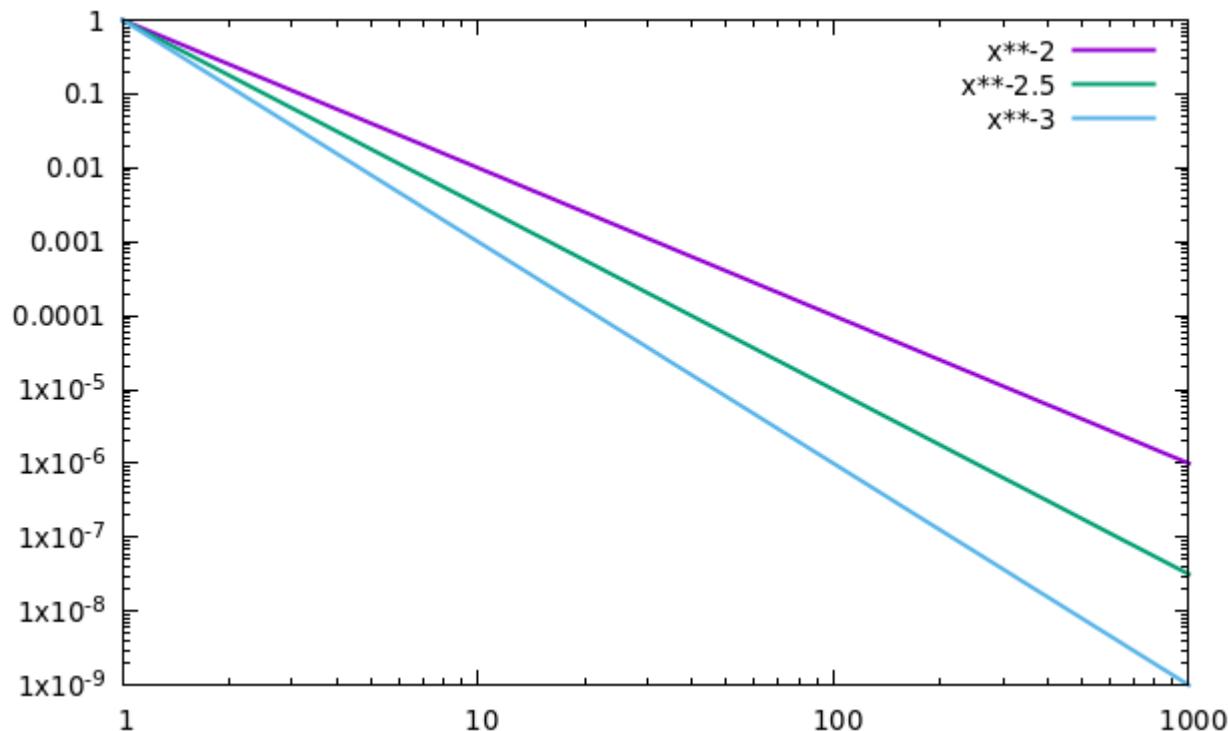
Above a certain x value,
the power law is
always higher than
the exponential

gnuplot

plot [4:20] 1.5**-x, x**-1.5, x**-2

Intermezzo: power-law “slope”

- Power-Law: $P(k) \propto k^{-\alpha}$



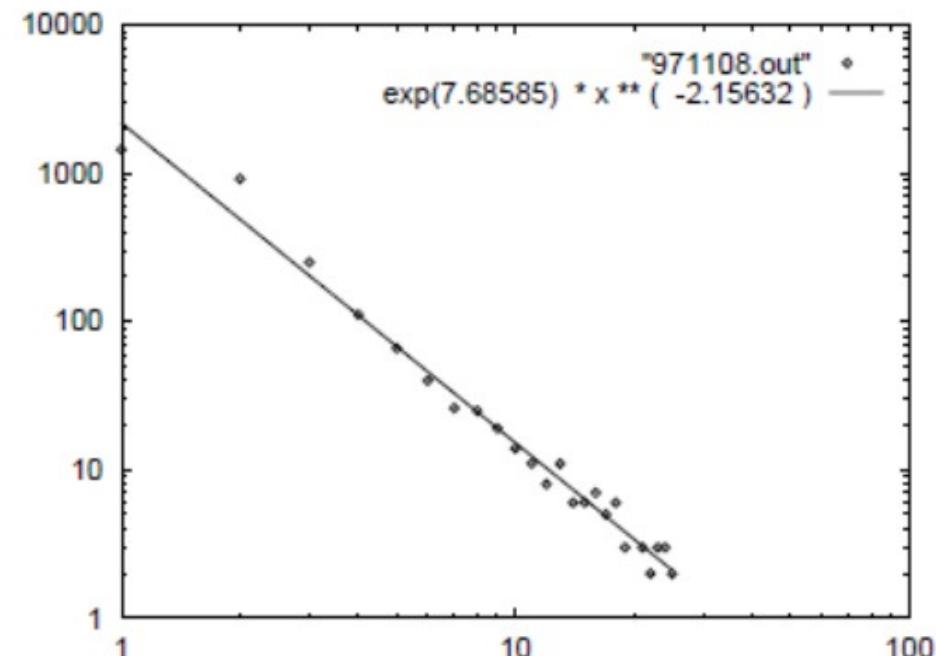
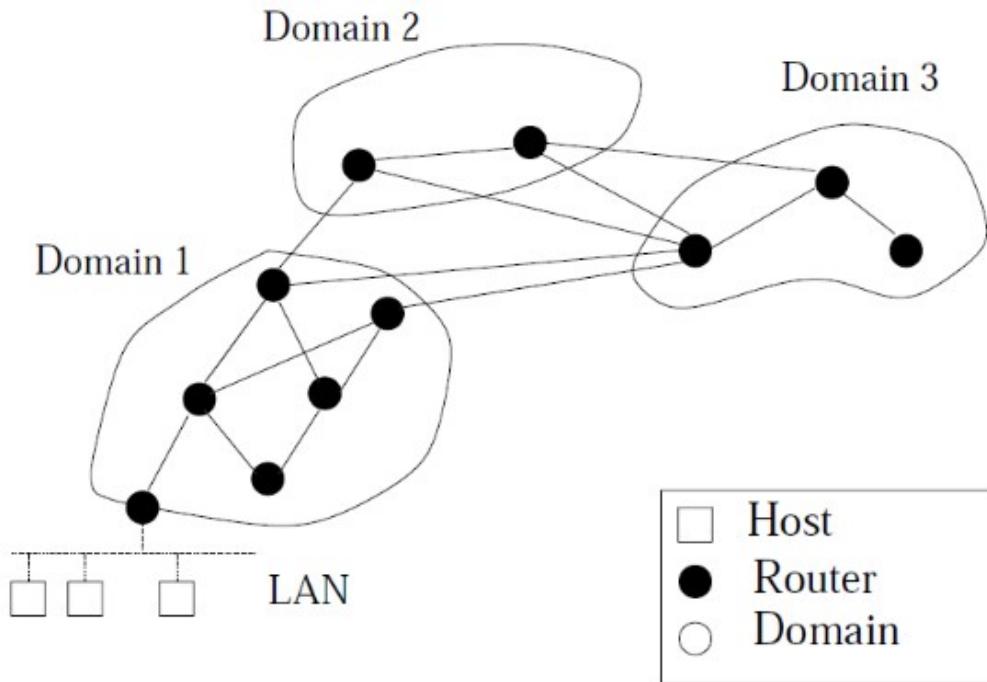
lower alpha (α)
will mean less
pronounced slope

gnuplot

```
plot [1:1000] x**-2 lw 2, x**-2.5 lw 2, x**-3 lw 2
```

Example: Internet Autonomous Systems

- First observed in Internet Autonomous Systems
[Faloutsos, Faloutsos and Faloutsos, 1999]



Internet domain topology

On Power-Law Relationships of the Internet Topology

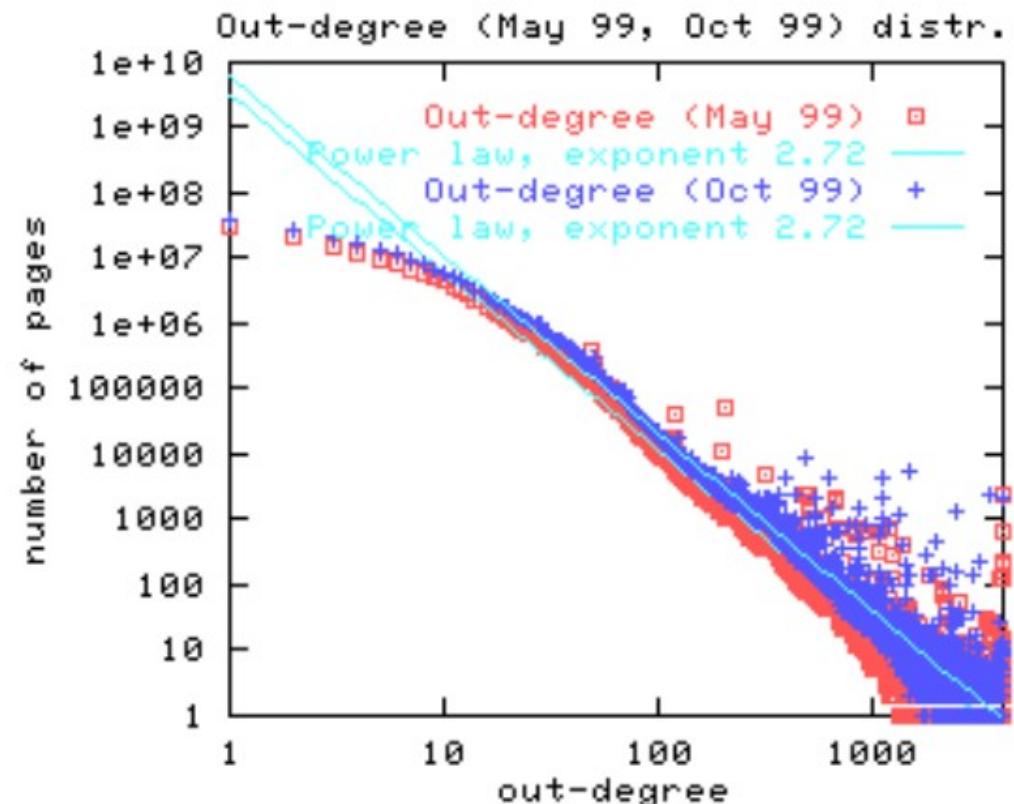
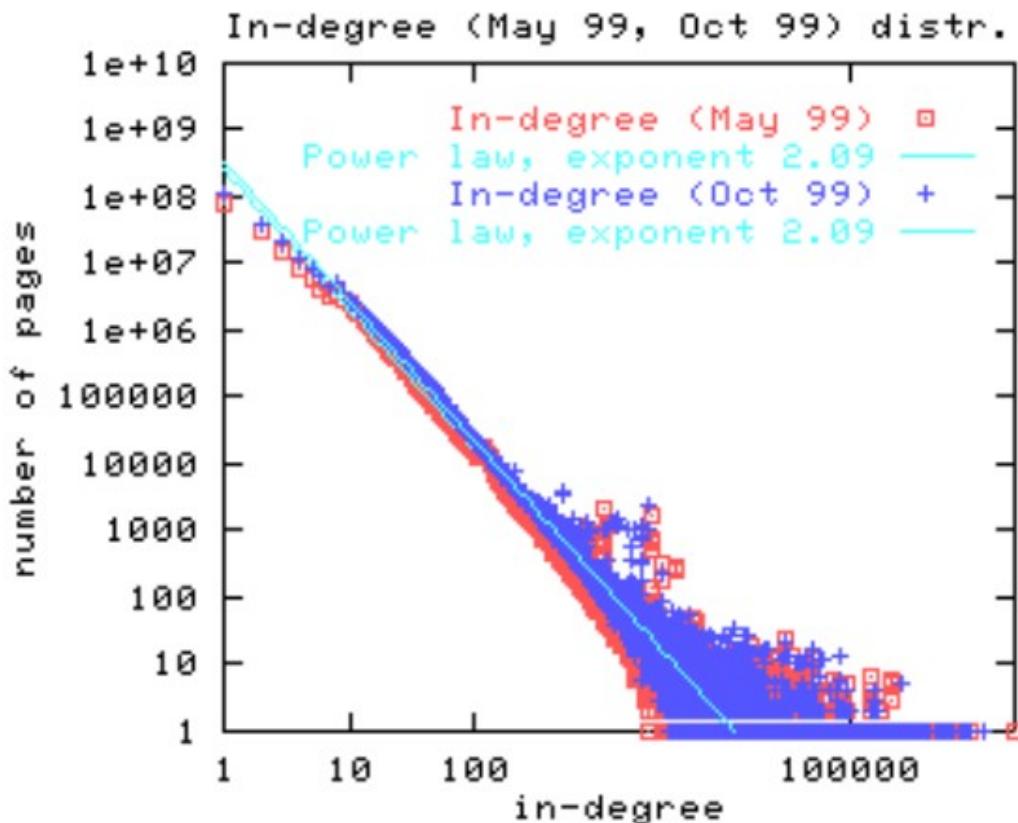
Michalis Faloutsos
U.C. Riverside
Dept. of Comp. Science
michalis@cs.ucr.edu

Petros Faloutsos
U. of Toronto
Dept. of Comp. Science
pfal@cs.toronto.edu

Christos Faloutsos *
Carnegie Mellon Univ.
Dept. of Comp. Science
christos@cs.cmu.edu

Example: World Wide Web

[Broder et al., 2000]



Graph structure in the Web

Andrei Broder^a, Ravi Kumar^{b,*}, Farzin Maghoul^a, Prabhakar Raghavan^b,
Sridhar Rajagopalan^b, Raymie Stata^c, Andrew Tomkins^b, Janet Wiener^c

^a AltaVista Company, San Mateo, CA, USA

^b IBM Almaden Research Center, San Jose, CA, USA

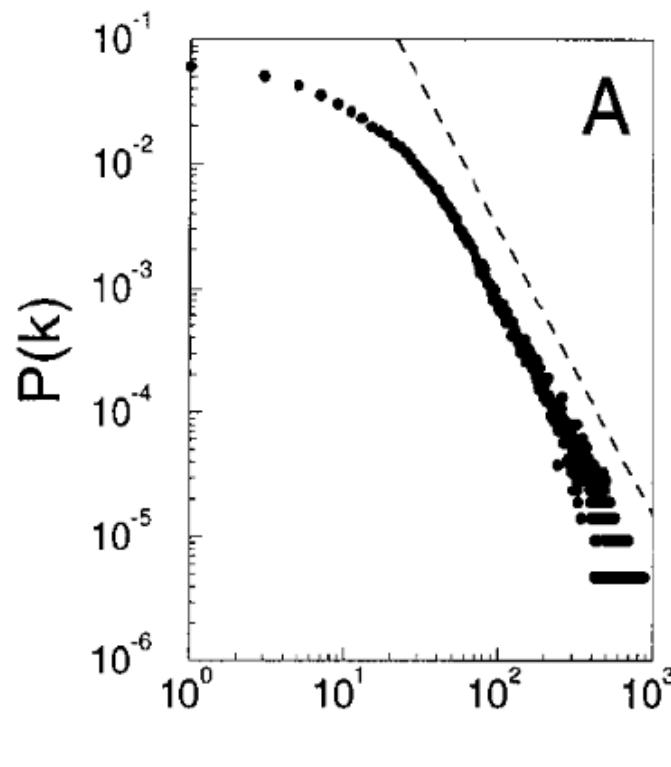
^c Compaq Systems Research Center, Palo Alto, CA, USA

Other Examples

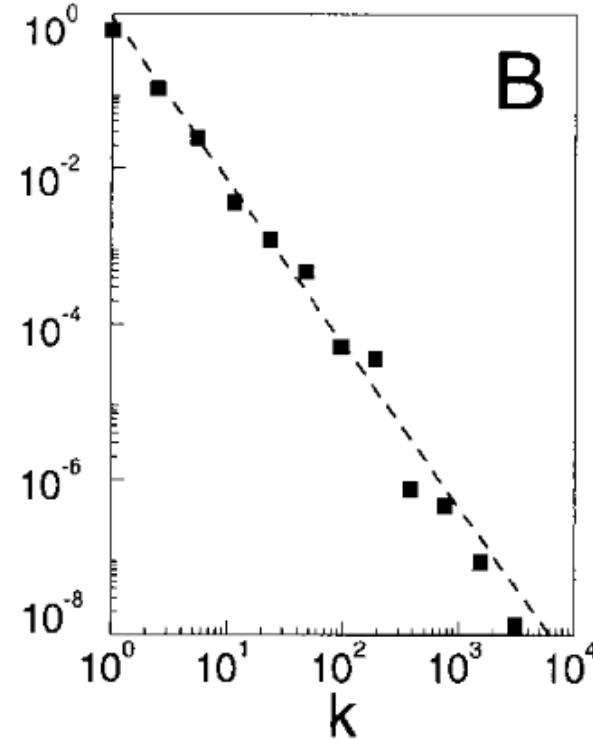
[Barabasi-Albert, 1999]

Emergence of Scaling in Random Networks

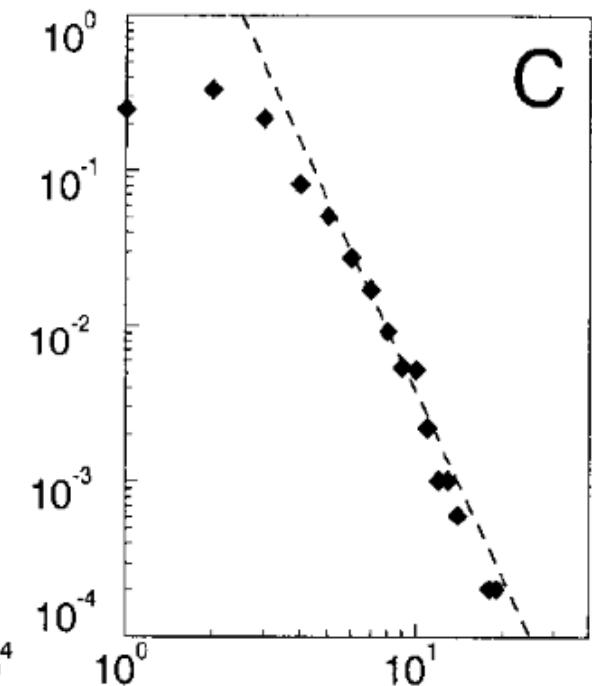
Albert-László Barabási* and Réka Albert



Actor collaborations

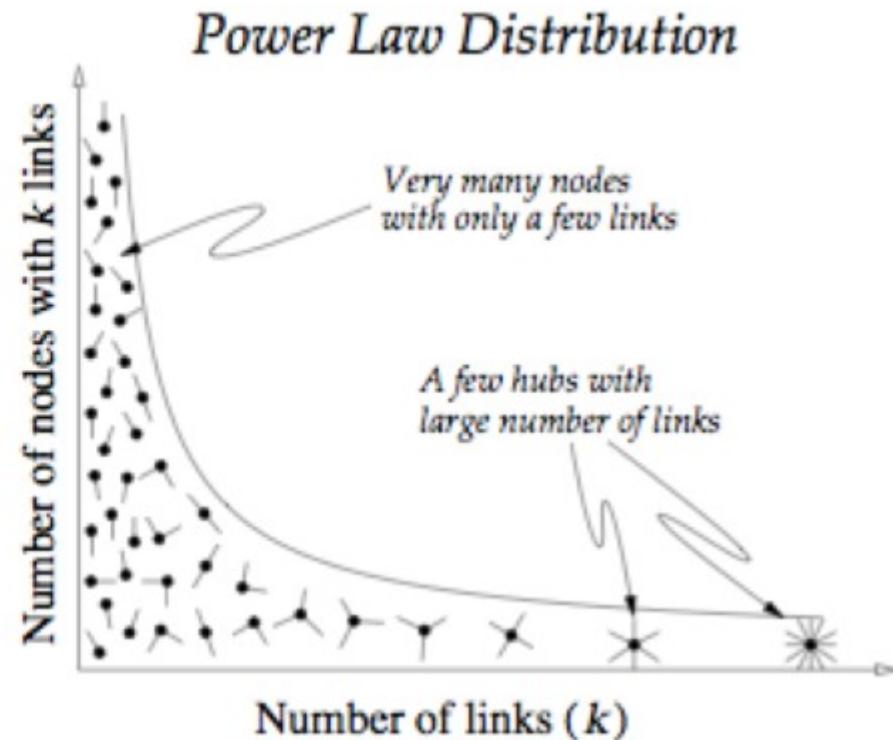
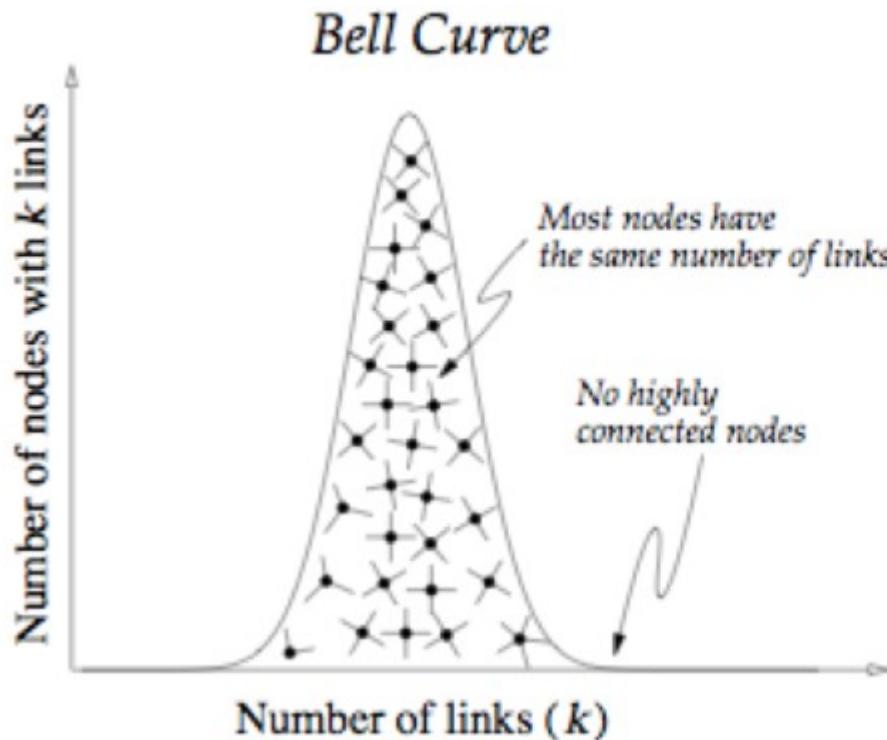


Web graph



Power-grid

Interpreting Power-Laws



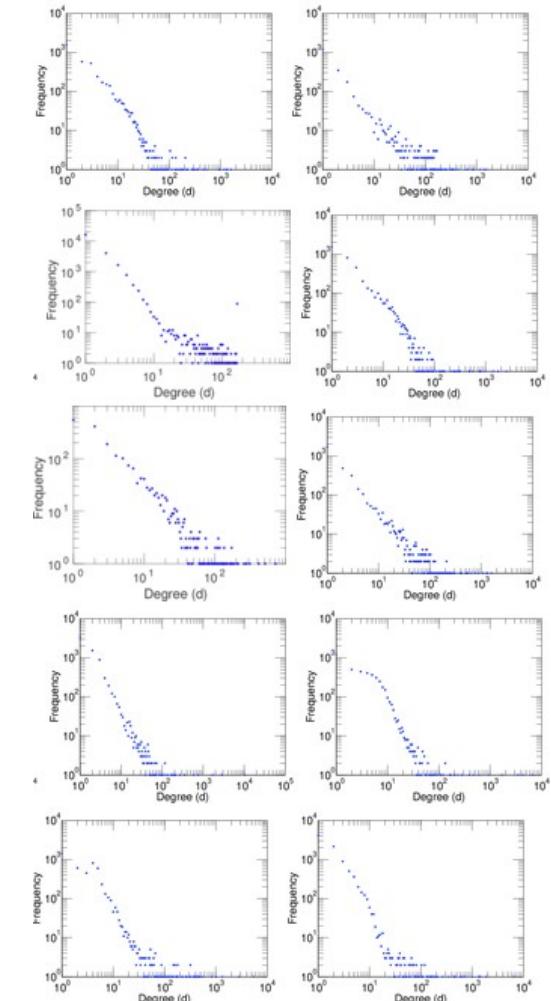
Power-Law Degree Exponent

- Power-law degree exponent is typically:

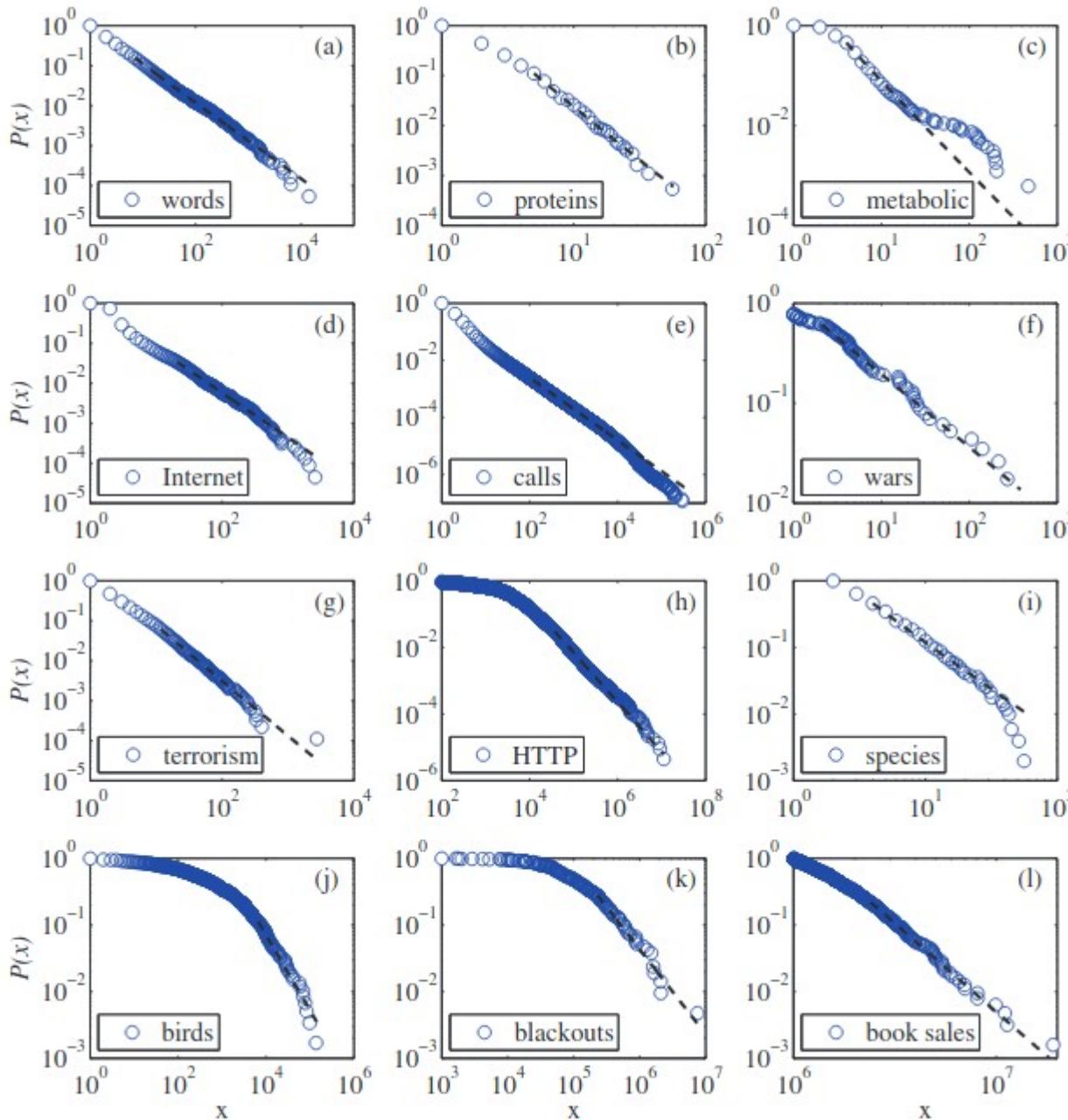
$$2 < \alpha < 3$$

- Examples

- Web graph:
 - $\alpha_{\text{in}} = 2.1$, $\alpha_{\text{out}} = 2.4$ [Broder et al. 00]
- Autonomous systems:
 - $\alpha = 2.4$ [Faloutsos 3 , 99]
- Actor-collaborations:
 - $\alpha = 2.3$ [Barabasi-Albert 00]
- Citations to papers:
 - $\alpha \approx 3$ [Redner 98]
- Online social networks:
 - $\alpha \approx 2$ [Leskovec et al. 07]



Power Laws are Everywhere

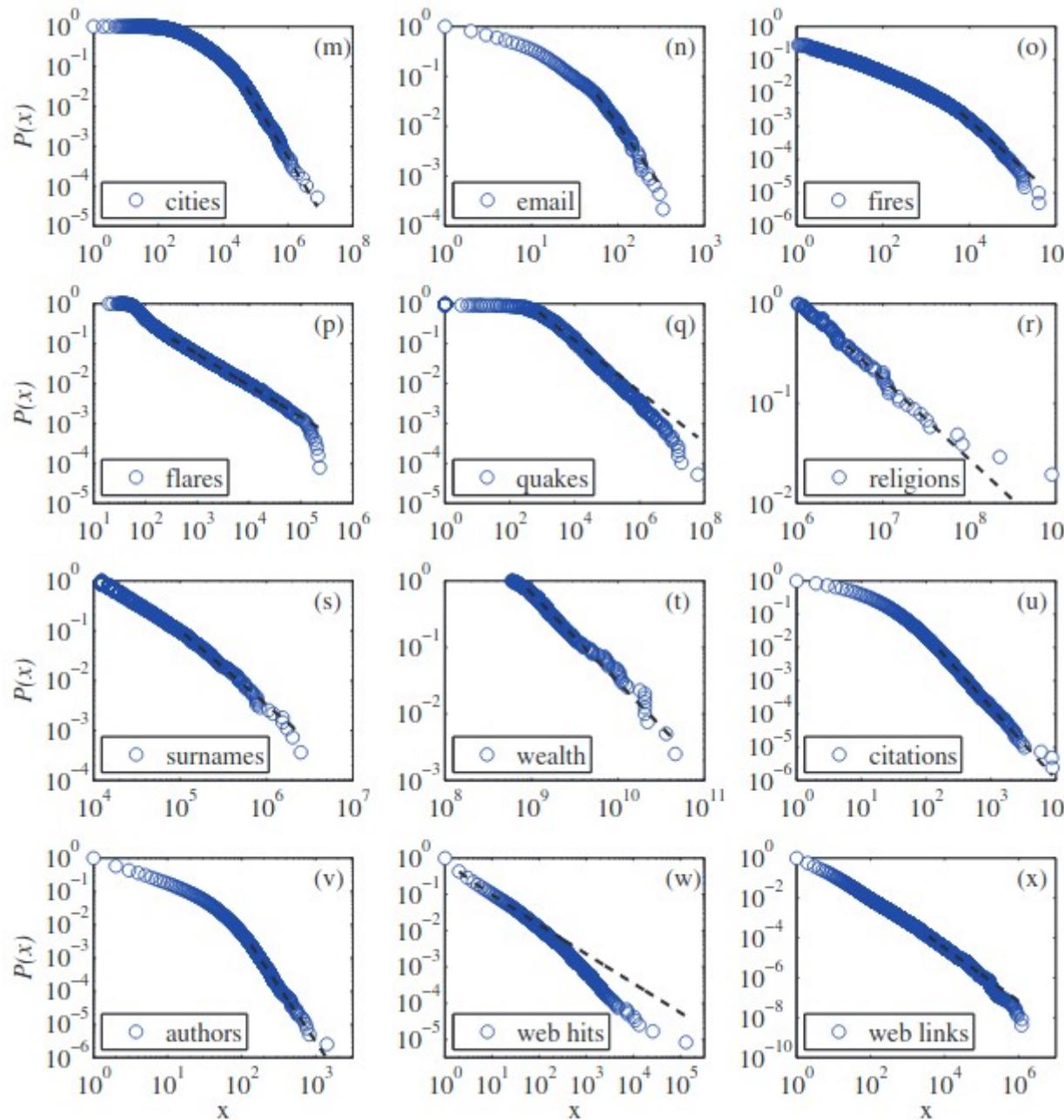


Power-Law Distributions in Empirical Data*

Aaron Clauset[†]
Cosma Rohilla Shalizi[‡]
M. E. J. Newman[§]

[Clauset, Shalizi, Newman, 2009]

Power Laws are Everywhere



Power-Law Distributions in Empirical Data*

Aaron Clauset[†]
Cosma Rohilla Shalizi[‡]
M. E. J. Newman[§]

[Clauset, Shalizi, Newman, 2009]

Not everyone likes Power Laws 😊

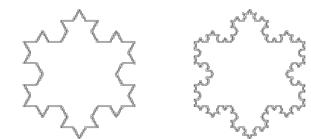


CMU grad-students at
the G20 meeting in
Pittsburgh in Sept 2009



Scale Free Networks

- Networks with a **power-law** tail in their degree distribution are often called **“scale-free networks”**
- Where does the term scale-free come from?
 - **Scale invariance:** there is no characteristic scale
 - means laws do not change if scales of length, energy, or other variables, are multiplied by a common factor
 - **Scale free function:** $f(\lambda x) = C(\lambda) f(x) \propto f(x)$
 - Power-law: $f(x) = ax^{-\alpha}$
 $f(\lambda x) = a(\lambda x)^{-\alpha} = \lambda^{-\alpha}(ax^{-\alpha}) = \lambda^{-\alpha} f(x) \propto f(x)$



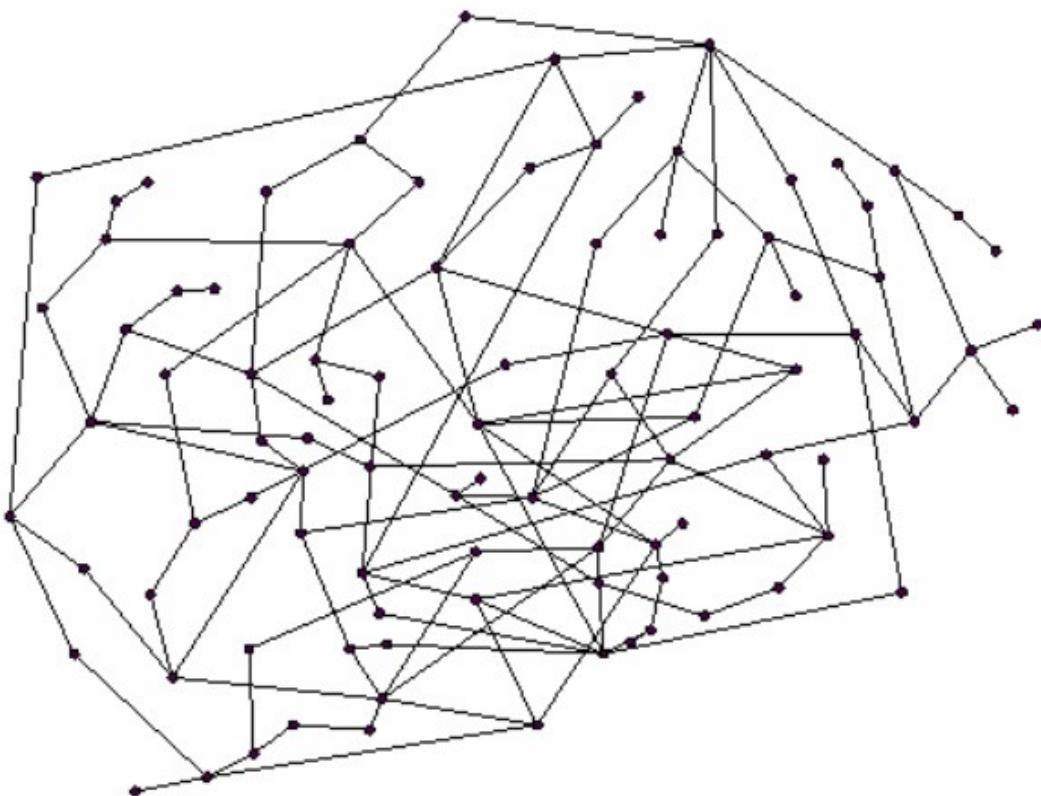
$C(\lambda)$ depends
only on λ

Log() or Exp() are not scale free

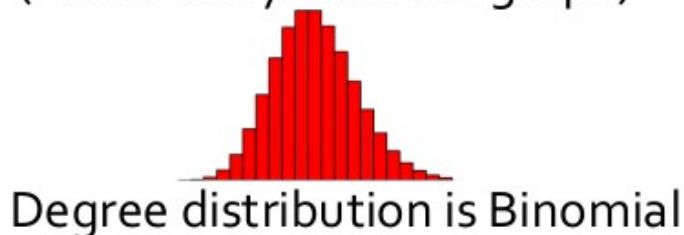
$$f(\lambda x) = \log(\lambda x) = \log(\lambda) + \log(x) = \log(\lambda) + f(x)$$

$$f(\lambda x) = \exp(\lambda x) = \exp(x^\lambda) = f(x)^\lambda$$

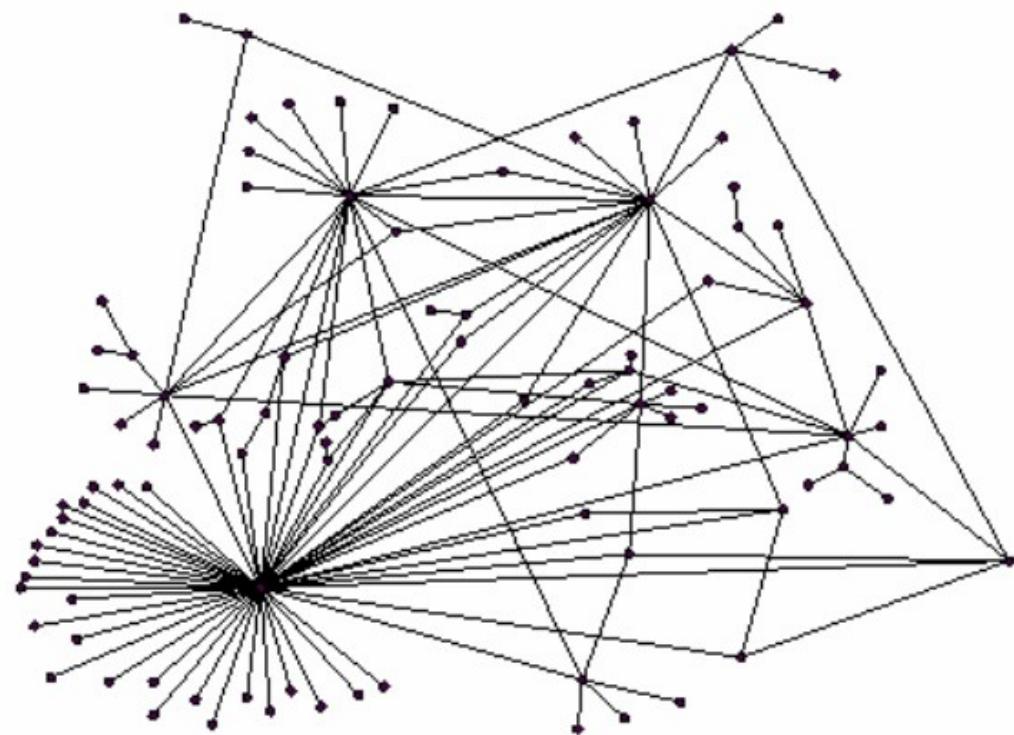
Random vs Scale Free



Random network
(Erdos-Renyi random graph)



Degree distribution is Binomial



Scale-free (power-law) network



Preferential Attachment Model

Rich Get Richer

- **New nodes are more likely to link to nodes that already have high degree**
- Herbert Simon's result:
 - Power-laws arise from “*Rich get richer*” (cumulative advantage)
- Examples:
 - **Citations** [de Solla Price '65]: New citations to a paper are proportional to the number it already has
 - Herding: If a lot of people cite a paper, then it must be good, and therefore I should cite it too
 - **Sociology: Matthew effect** (http://en.wikipedia.org/wiki/Matthew_effect)
 - “For whoever has will be given more, and they will have an abundance. Whoever does not have, even what they have will be taken from them.”
 - Eminent scientists often get more credit than a comparatively unknown researcher, even if their work is similar

ON A CLASS OF SKEW DISTRIBUTION FUNCTIONS

BY HERBERT A. SIMON†
Carnegie Institute of Technology

Networks of Scientific Papers

The pattern of bibliographic references indicates the nature of the scientific research front.

Derek J. de Solla Price

Model: Preferential Attachment

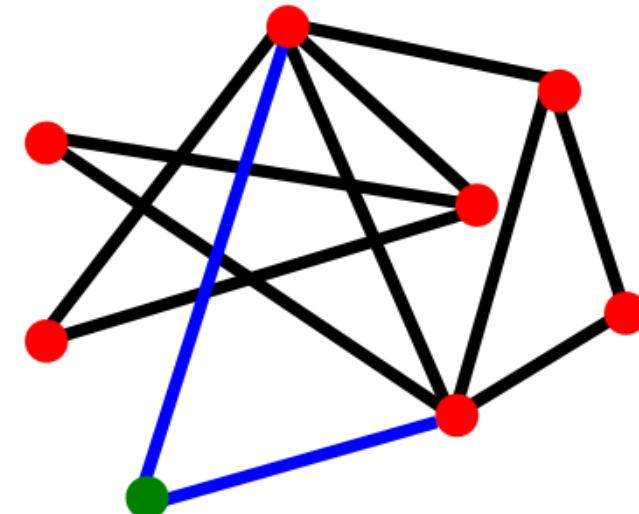
- **Preferential attachment:**

[Barabasi-Albert '99] **(Barabasi-Albert model)**

- Nodes arrive in order **1,2,...,n**
- At step j , let d_i be the degree of a previous node i
- A new node j arrives and creates **m out-links**
- Probability of j linking to a previous node i is proportional to degree **d_i of node i**

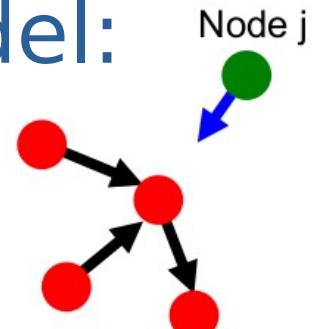
Emergence of Scaling in
Random Networks
Albert-László Barabási* and Réka Albert

$$P(j \rightarrow i) = \frac{d_i}{\sum_k d_k}$$



Results for Simple Model

- We analyze the following **simple** model:
 - Nodes arrive in order $1, 2, 3, \dots, n$
 - When *node j* is created it makes a **single out-link** to an earlier node *i* chosen:
 - 1) With prob. p , *j* links to *i* chosen **uniformly at random** (from among all earlier nodes)
 - 2) With prob. $1 - p$, node *j* chooses *i* uniformly at random & links **to a random node v that i points to**
 - **This is same as saying:** With prob. $1 - p$, node *j* links to node *v* with prob. proportional to d_v (the in-degree of *v*)
 - Our graph is **directed**: every node has out-degree 1



Results for Simple Model

- **Claim:** The described model generates networks where the fraction of nodes with **in-degree k** scales as:

$$P(d_i = k) \propto k^{-(1 + \frac{1}{q})}$$

where $q=1-p$

So we get power-law degree distribution with exponent:

$$\alpha = 1 + \frac{1}{1-p}$$

The model gives a **power-law**

Preferential Attachment: The Good

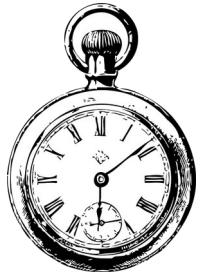
- Preferential attachment gives **power-law** in-degrees!
- Intuitively reasonable process
- Can **tune** model parameter p to get the observed exponent
 - On the web, $P[\text{node has in-degree } k] \sim k^{-2.1}$
 - $2.1 = 1 + 1/(1-p) \rightarrow p \sim 0.1$

$$p = 0 \rightarrow P(d_i = k) \sim k^{-2}$$

$$p = 0.5 \rightarrow P(d_i = k) \sim k^{-3}$$

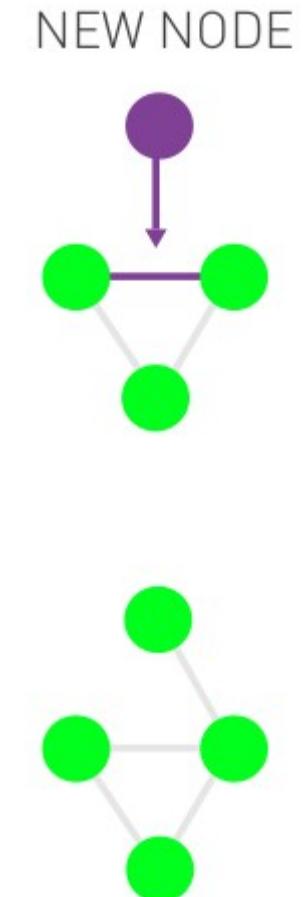
Preferential Attachment: The Bad

- Preferential attachment is **not so good at predicting network structure**
 - **Age-degree correlation**
 - Node degree is proportional to its age
 - Possible Solution: Node fitness (virtual degree)
 - **Links among high degree nodes:**
 - On the web nodes sometimes avoid linking to each other
- **Further questions:**
 - What is a reasonable model for **how people sample network nodes and link to them?**



Origins of Preferential Attachment

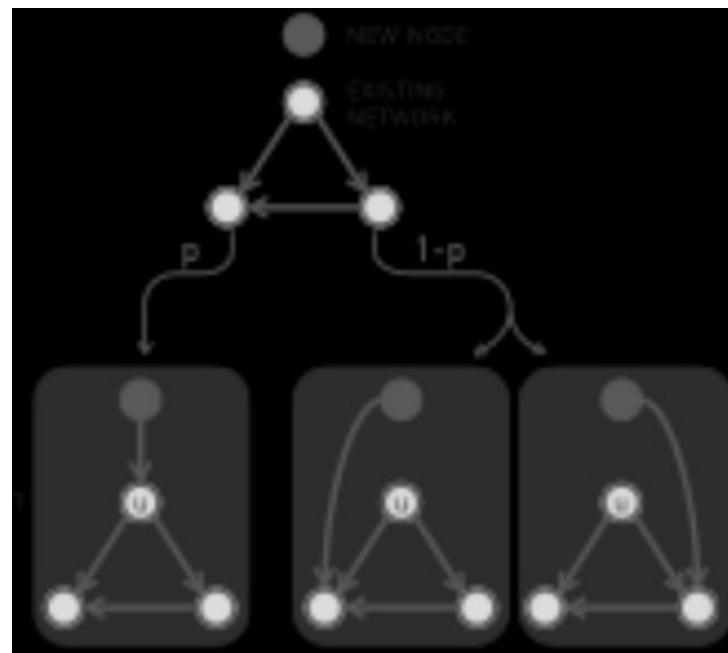
- **Link Selection Model:** perhaps the simplest example of a local or random mechanism capable of generating preferential attachment
 - **Growth:** At each time step we add a new node to the network
 - **Link selection:** We select a link at random and connect the new node to one of the nodes at the two ends of the selected link
- This simple mechanism generates **preferential attachment**
 - Why? Because nodes are picked with probability proportional to their number of edges



Origins of Preferential Attachment

- **Copying Model:**

- (a) **Random Connection:** with prob. p the new node links to random node v
- (b) **Copying:** With prob. $1 - p$ randomly choose an outgoing link of node v and connect the new node to the selected link's target
 - The new node “copies” one of the links of an earlier node



Origins of Preferential Attachment

- Analysis of the **copying model**:
 - (a) the probability of selecting a node is $1/N$
 - (b) is equivalent to selecting a node linked to a randomly selected link. The probability of selecting a degree- k node through the copying process of step (b) is $k/2E$ for undirected networks
 - Again, the likelihood that the new node will connect to a degree- k node follows preferential attachment
- Examples:
 - **Social networks:** Copy your friend's friends.
 - **Citation Networks:** Copy references from papers we read
 - **Protein interaction networks:** gene duplication

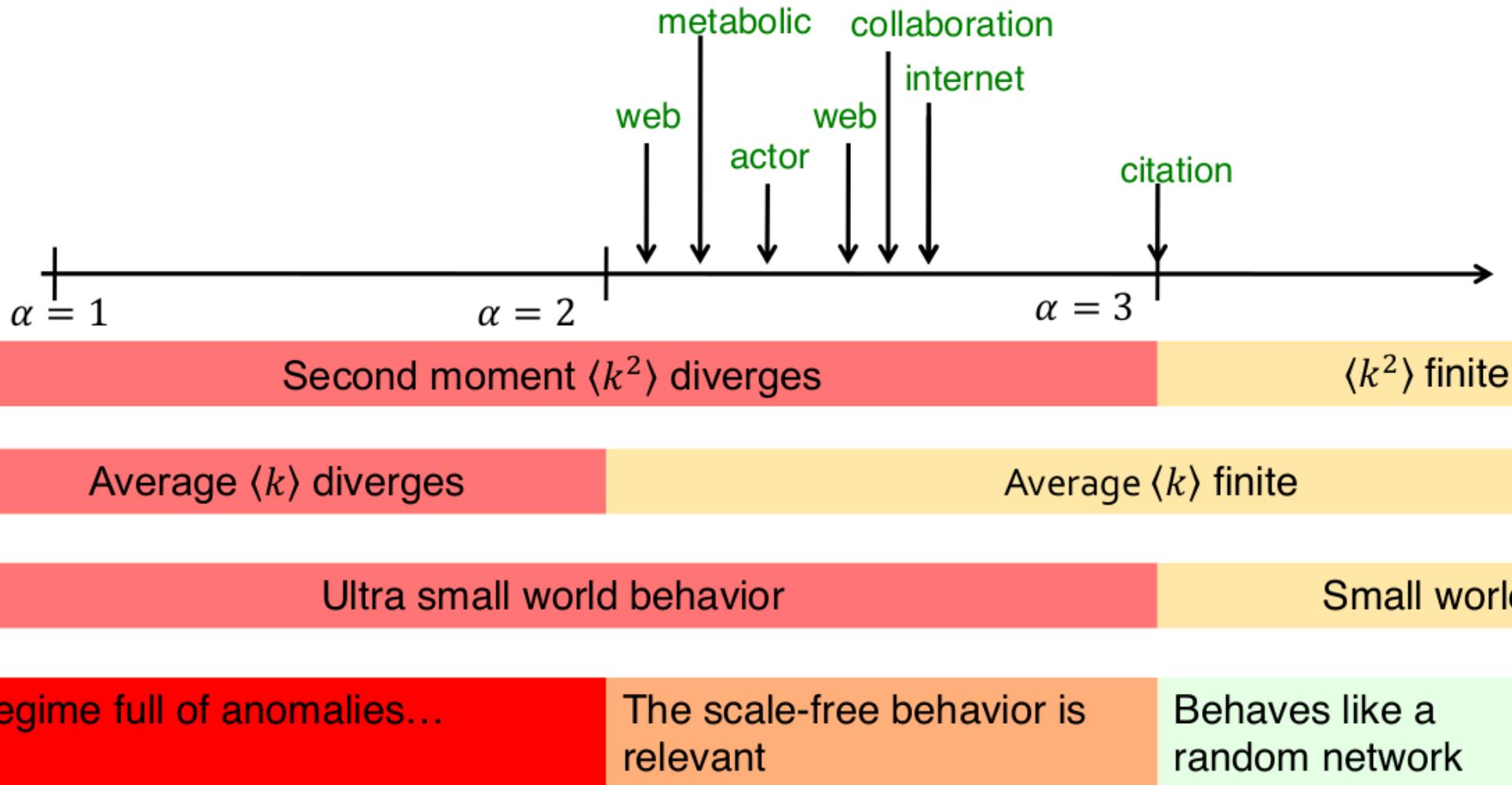
Many models lead to power-laws

- **Copying mechanism** (directed network)
 - Select a node and an edge of this node
 - Attach to the endpoint of this edge
- **Walking on a network** (directed network)
 - The new node connects to a node, then to every first, second, ... neighbor of this node
- **Attaching to edges**
 - Select an edge and attach to both endpoints of this edge
- **Node duplication**
 - Duplicate a node with all its edges
 - Randomly prune edges of new node

Distances in Preferential Attachment

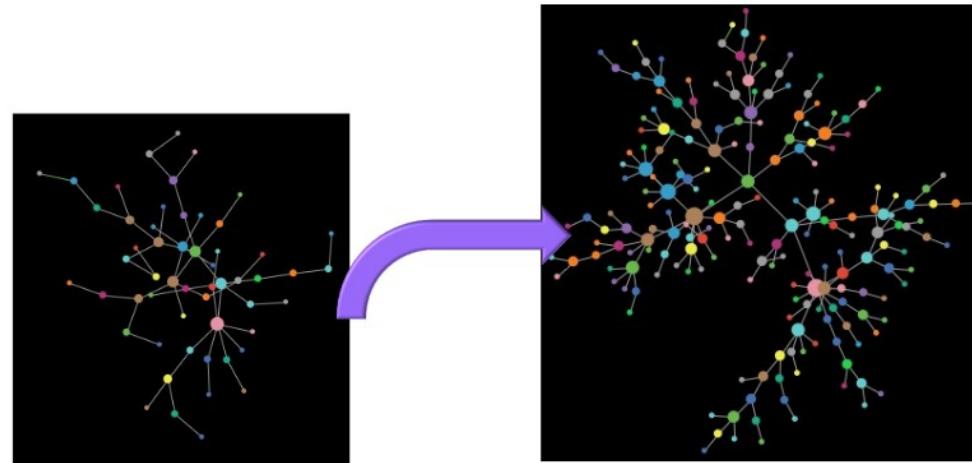
\bar{h} =	α	Size of the biggest hub is of order $O(N)$. Most nodes can be connected within two steps, thus the average path length will be independent of the network size n .
Ultra small world	$const$	$\alpha = 2$
	$\frac{\log \log n}{\log(\alpha-1)}$	$2 < \alpha < 3$
	$\frac{\log n}{\log \log n}$	$\alpha = 3$
Small world	$\log n$	$\alpha > 3$
Avg. path length	Degree exponent	The second moment of the distribution is finite, thus in many ways the network behaves as a random network. Hence the average path length follows the result that we derived for the random network model earlier.

Scale-Free Networks: Overview



Scale-Free Networks: Ingredients

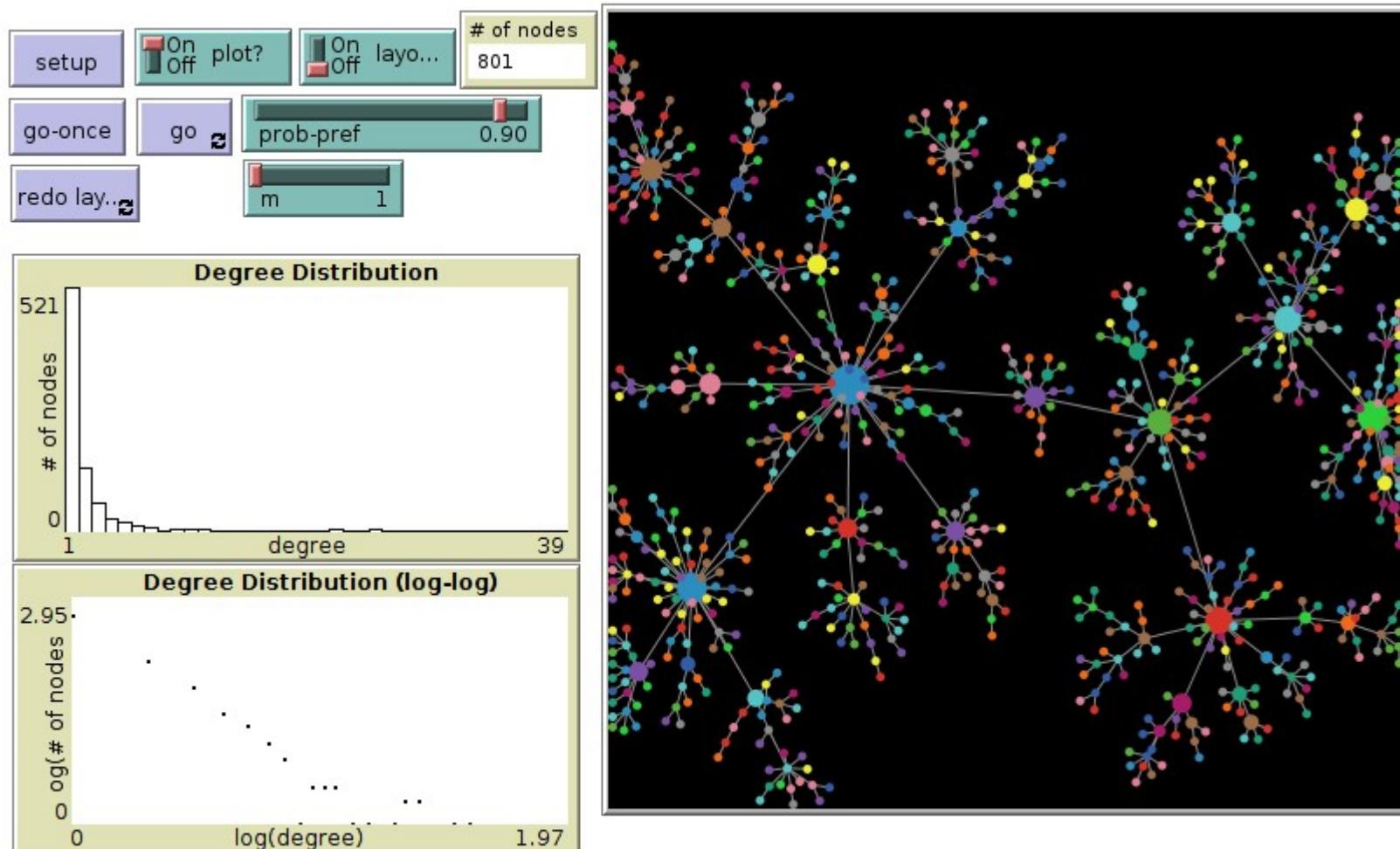
- Nodes appear over time (**growth**)



- Nodes prefer to attach to nodes with many connections (**preferential attachment, cumulative advantage**)



NetLogo: Preferential Attachment

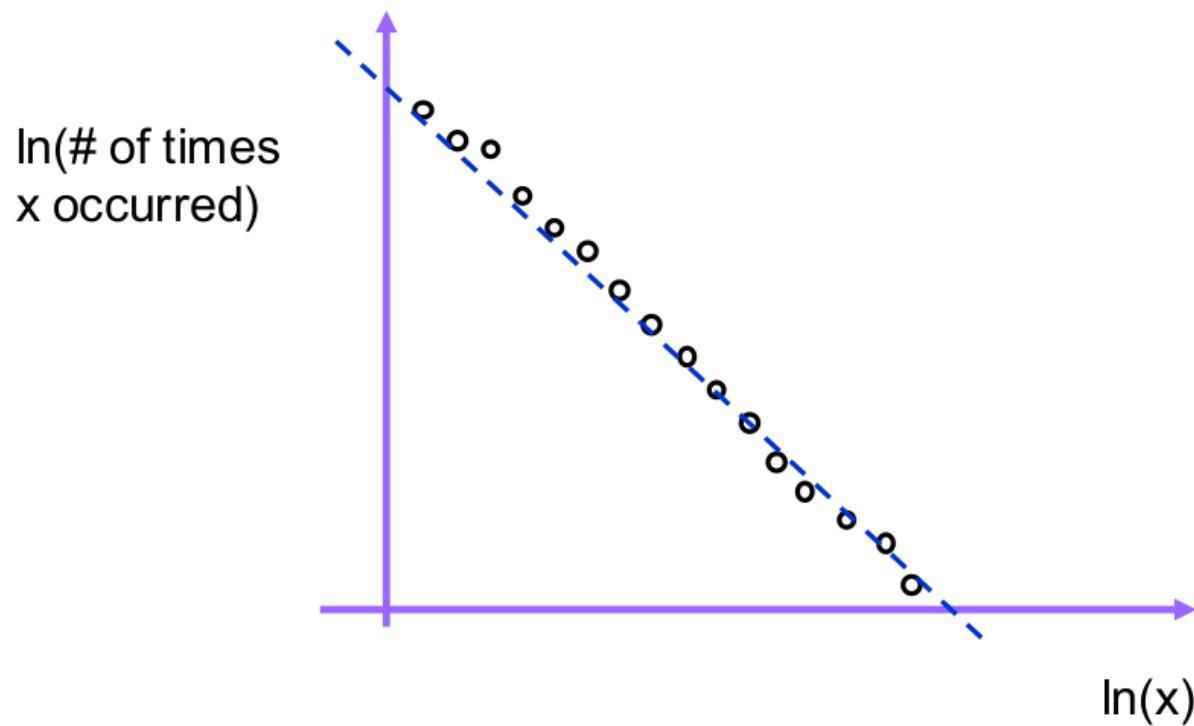


RAndPrefAttachment.nlogo

Fitting power-law distributions

Simple Binning

- Most common and not very accurate method:
 - Bin the different values of x and create a frequency histogram



$\ln(x)$ is the natural logarithm of x , but any other base of the logarithm will give the same exponent of α because $\log_{10}(x) = \ln(x)/\ln(10)$

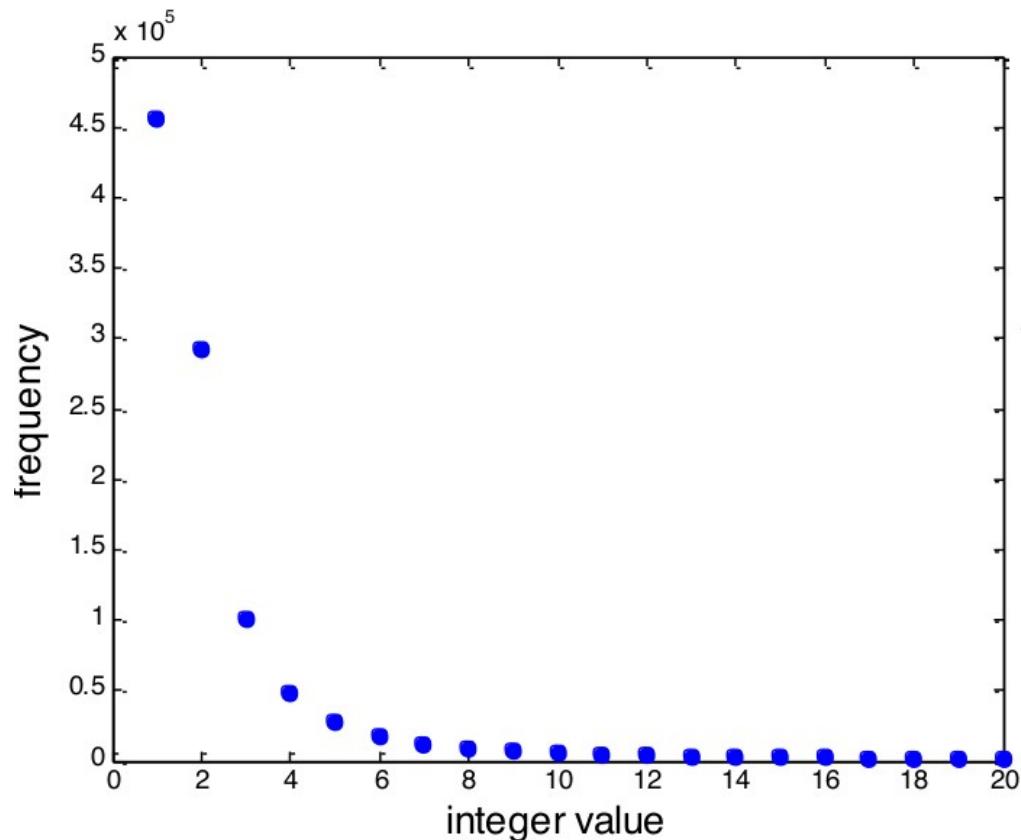
x can represent various quantities, the indegree of a node, the magnitude of an earthquake, the frequency of a word in text

Example on an artificially generated data set

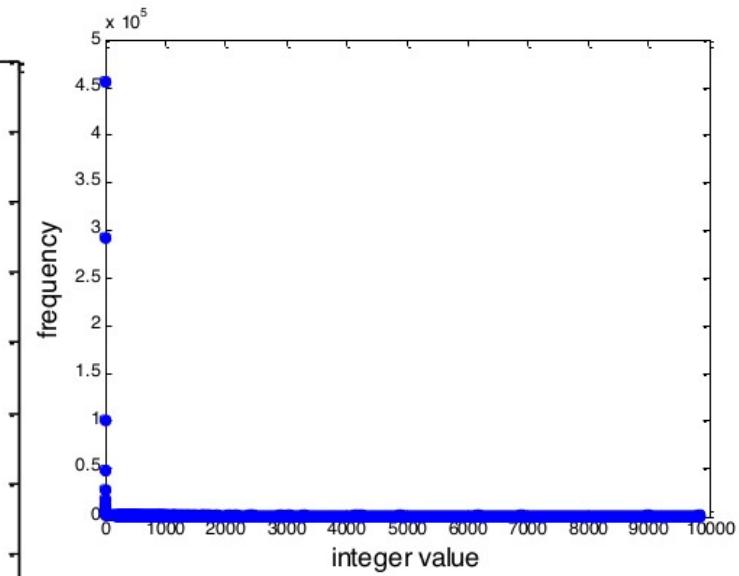
- Take 1 million random numbers from a distribution with $\alpha = 2.5$
- Can be generated using the so-called **“transformation method”**
- Generate random numbers r on the unit interval $0 \leq r < 1$
- Then $x = (1-r)^{-1/(\alpha-1)}$ is a **random power law** distributed real number in the range $1 \leq x < \infty$

Linear scale plot of simple bin. of the data

- Number of times 1 or 3843 or 99723 occurred
- Power-law relationship not as apparent
- Only makes sense to look at smallest bins



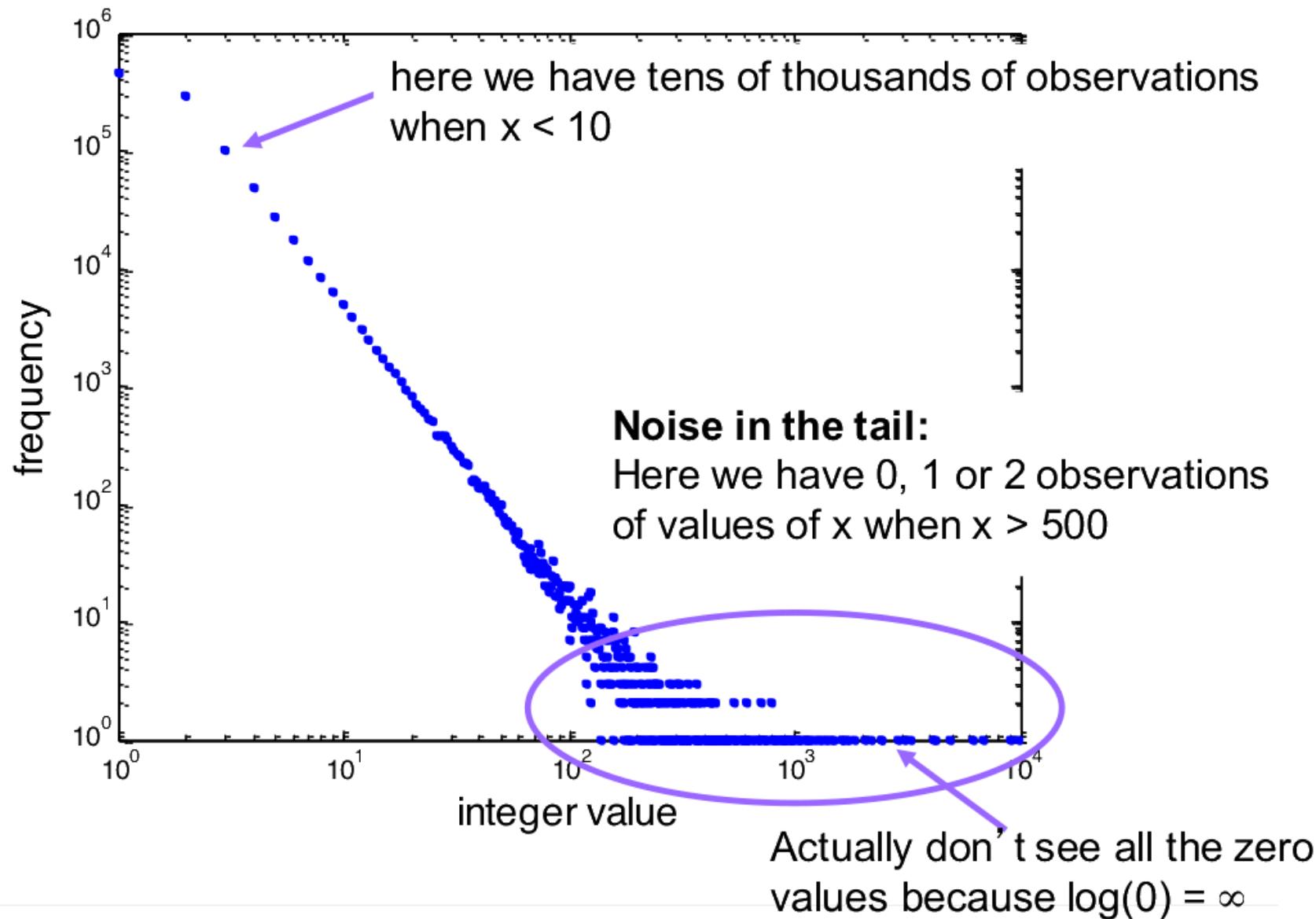
First few bins



Whole range

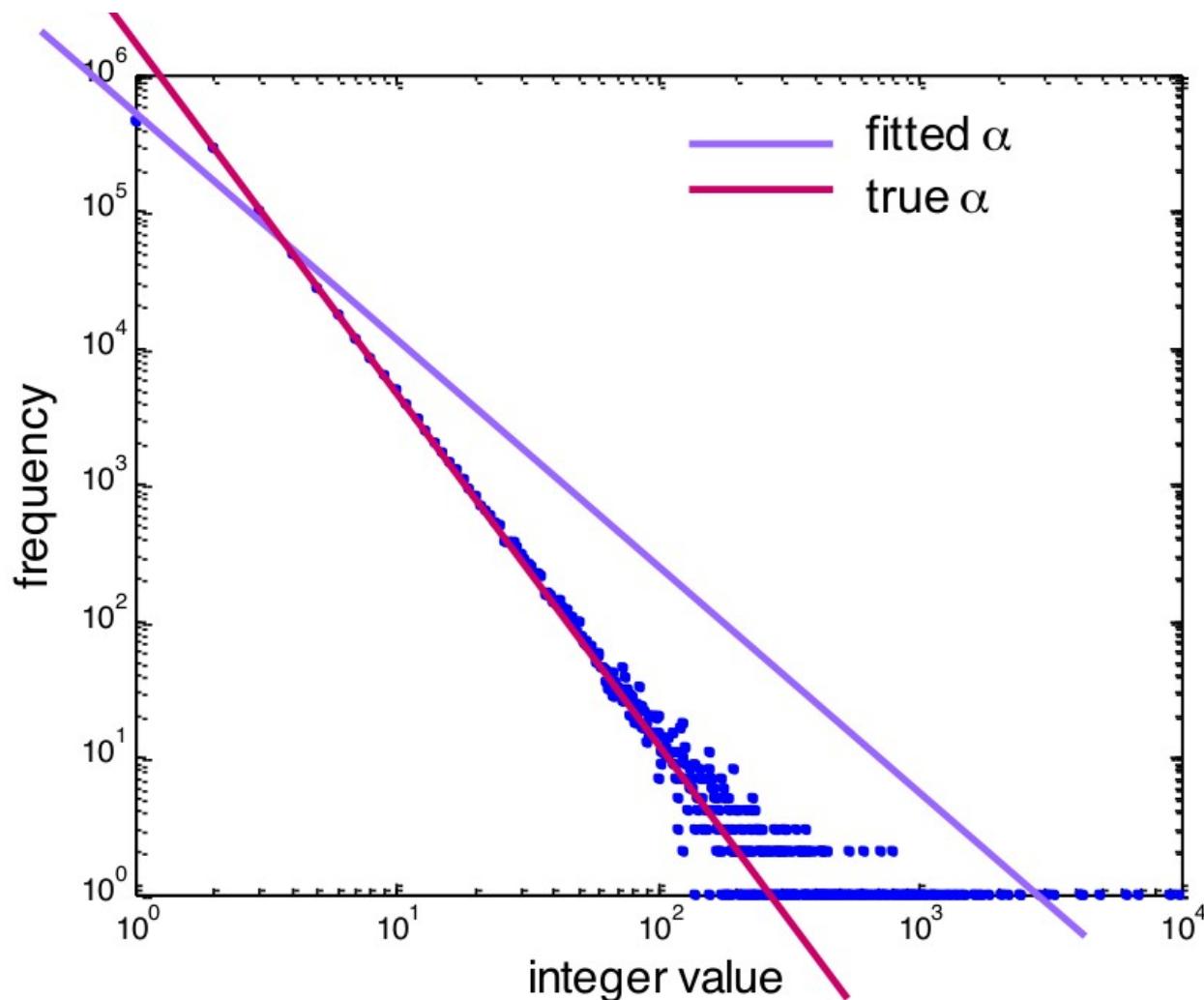
Log-log scale plot of simple bin. of the data

- Same bins, but plotted on a log-log scale



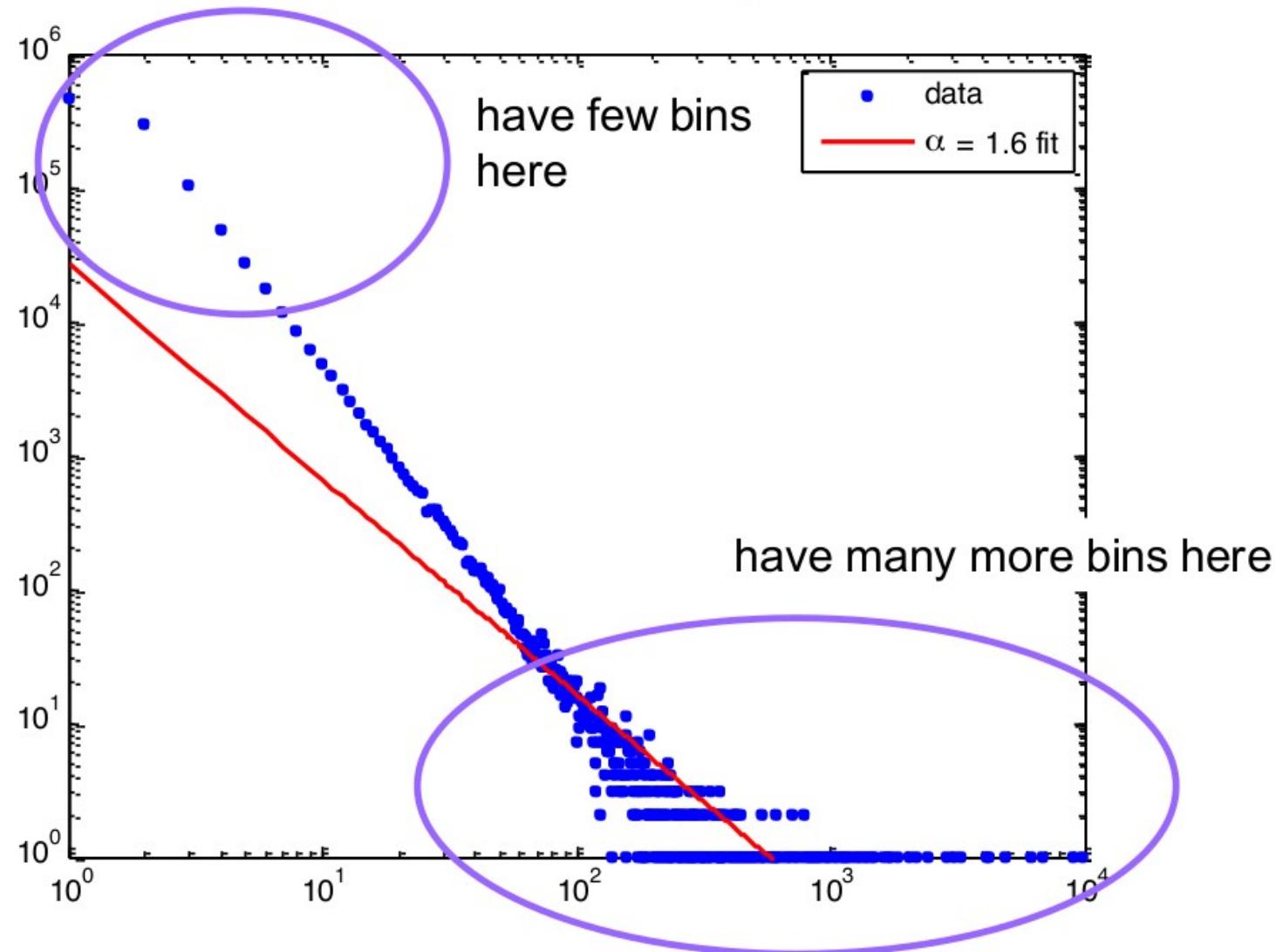
Log-log scale plot of simple bin. of the data

- Fitting a straight line to it via least squares regression will give values of the exponent α that are too low



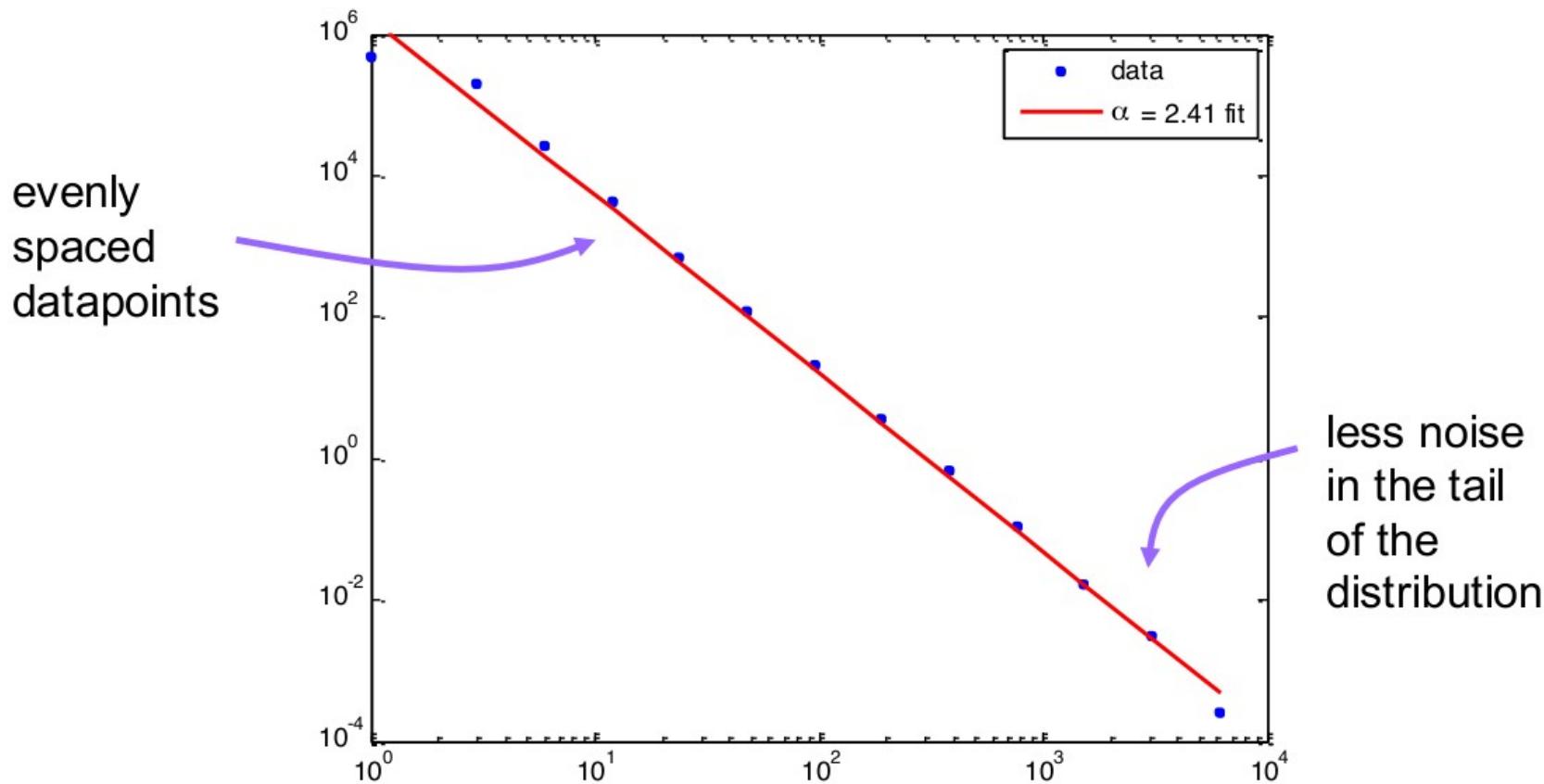
What goes wrong with simple binning

- Noise in the tail skews the regression result



First solution: logarithmic binning

- Bin data into **exponentially wider bins**:
 - 1, 2, 4, 8, 16, 32, ...
- **Normalize by the width of the bin**



- Disadvantage: binning smoothes out data but also loses information

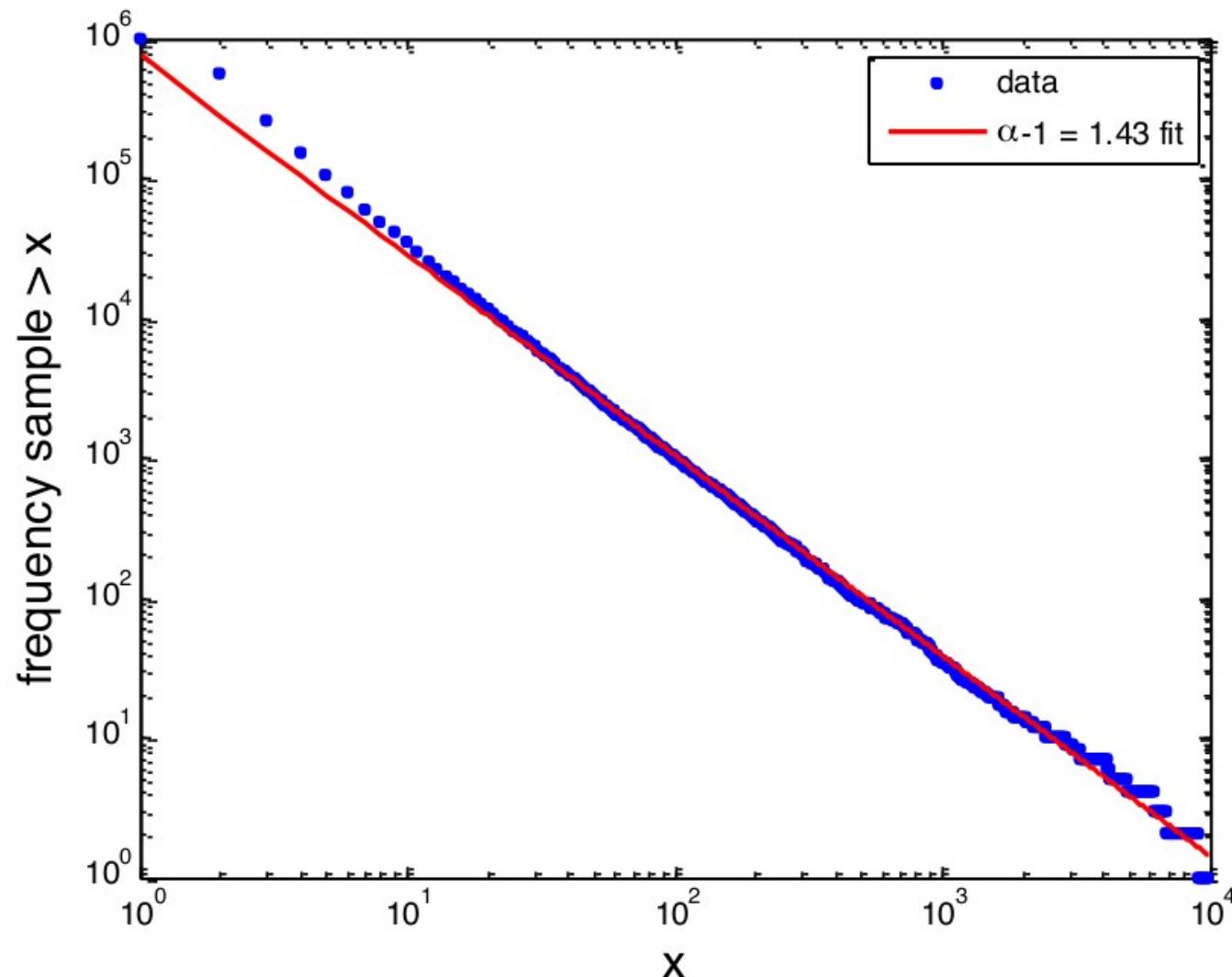
Second solution: cumulative binning

- No loss of information
 - No need to bin, has value at each observed value of x
- But now have **cumulative distribution**
 - i.e. how many of the values of x are at least X
- The **cumulative probability** of a power law probability distribution **is also a power law** but with an exponent $\alpha - 1$

$$\int cx^{-\alpha} = \frac{c}{1-\alpha} x^{-(\alpha-1)}$$

Fitting via regression to the cumulative distribution

- Fitted exponent (2.43) much closer to actual (2.5)

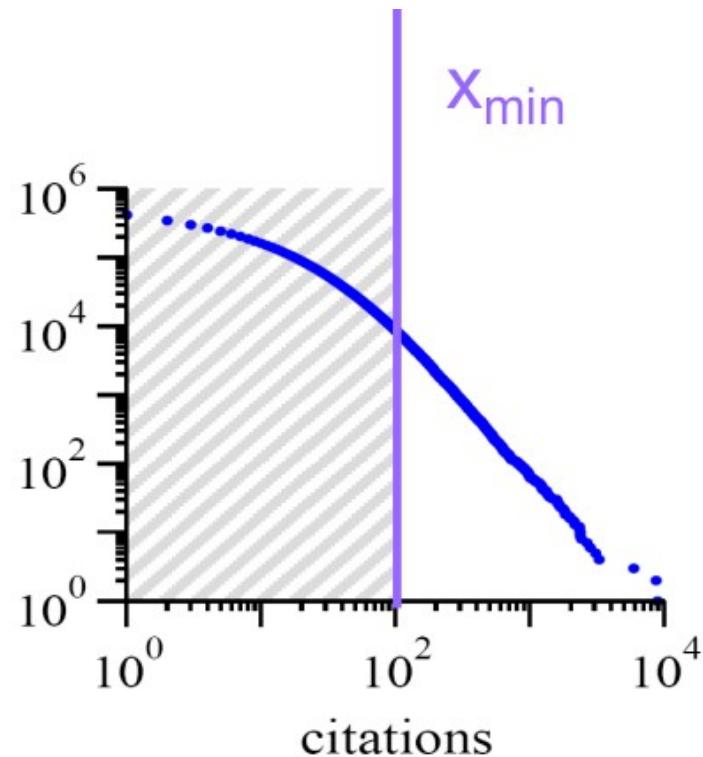


Where to start fitting?

- some data exhibit a power law
only in the tail
- after **binning or taking the cumulative distribution** you can fit to the tail
- so need to select an x_{min} the value of x where you think the power-law starts
- certainly x_{min} needs to be greater than 0 because $x^{-\alpha}$ is infinite at $x = 0$

Example of power-law in tail

- Distribution of citations to papers
- Power-law is evident only in the tail ($x_{min} > 100$ citations)



Power laws, Pareto distributions and Zipf's law

M.E.J. NEWMAN*

Maximum likelihood fitting - best

- You have to be sure you have a power-law distribution (this will just give you an exponent but not a goodness of fit)

$$\alpha = 1 + n \left[\sum_{i=1}^n \ln \frac{x_i}{x_{\min}} \right]^{-1}$$

- x_i are all your data points, and you have n of them
- for our data set we get $\alpha = 2.503$ - pretty close!

Some exponents for real world data

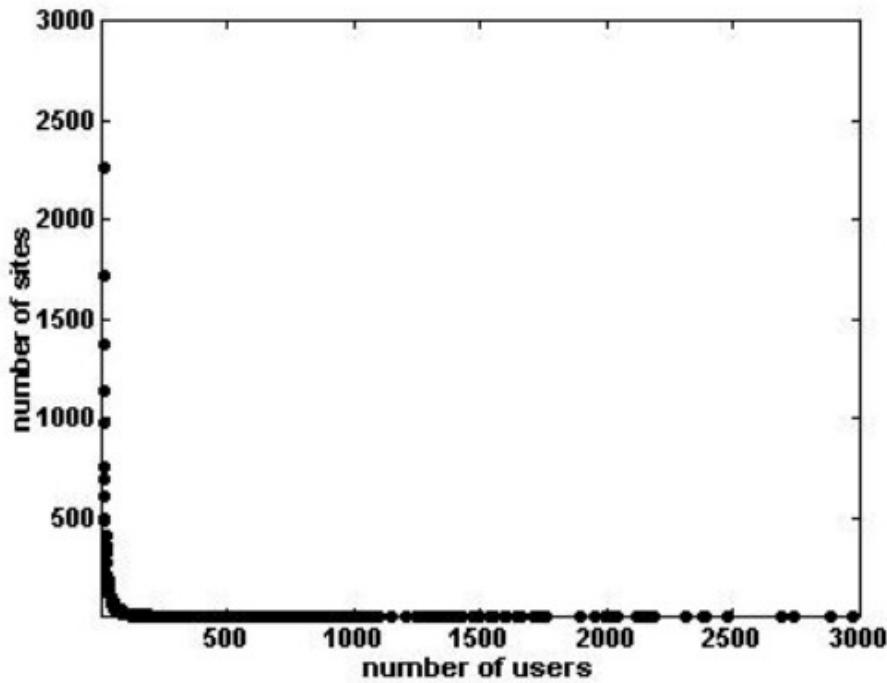
	x_{\min}	exponent α
frequency of use of words	1	2.20
number of citations to papers	100	3.04
number of hits on web sites	1	2.40
copies of books sold in the US	2 000 000	3.51
telephone calls received	10	2.22
magnitude of earthquakes	3.8	3.04
diameter of moon craters	0.01	3.14
intensity of solar flares	200	1.83
intensity of wars	3	1.80
net worth of Americans	\$600m	2.09
frequency of family names	10 000	1.94
population of US cities	40 000	2.30

Many real world networks are power-law

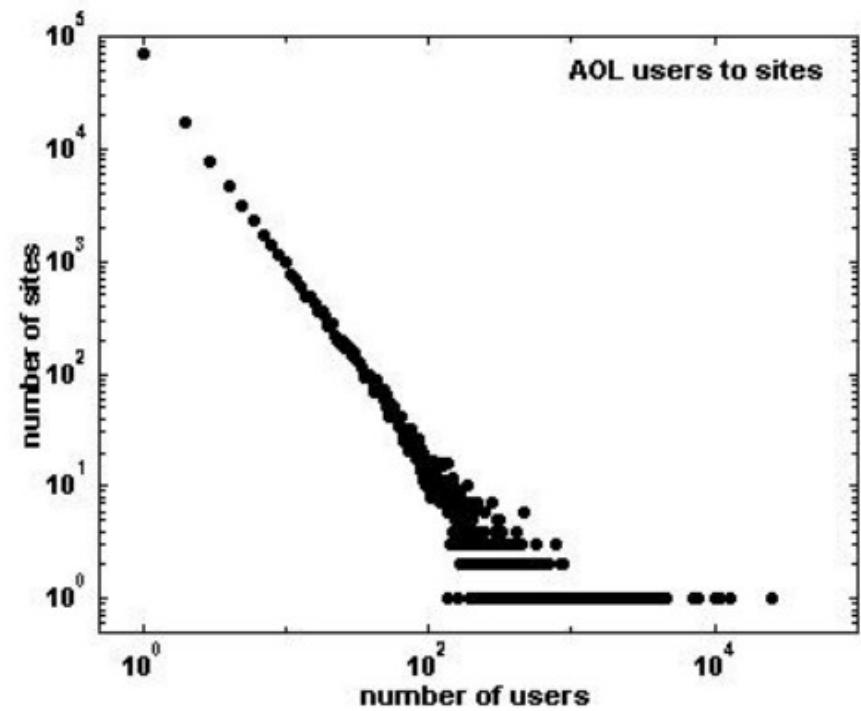
	exponent α (in/out degree)
film actors	2.3
telephone call graph	2.1
email networks	1.5/2.0
sexual contacts	3.2
WWW	2.3/2.7
internet	2.5
peer-to-peer	2.1
metabolic network	2.2
protein interactions	2.4

Example on a real data set

- Number of AOL visitors to different websites back in 1997



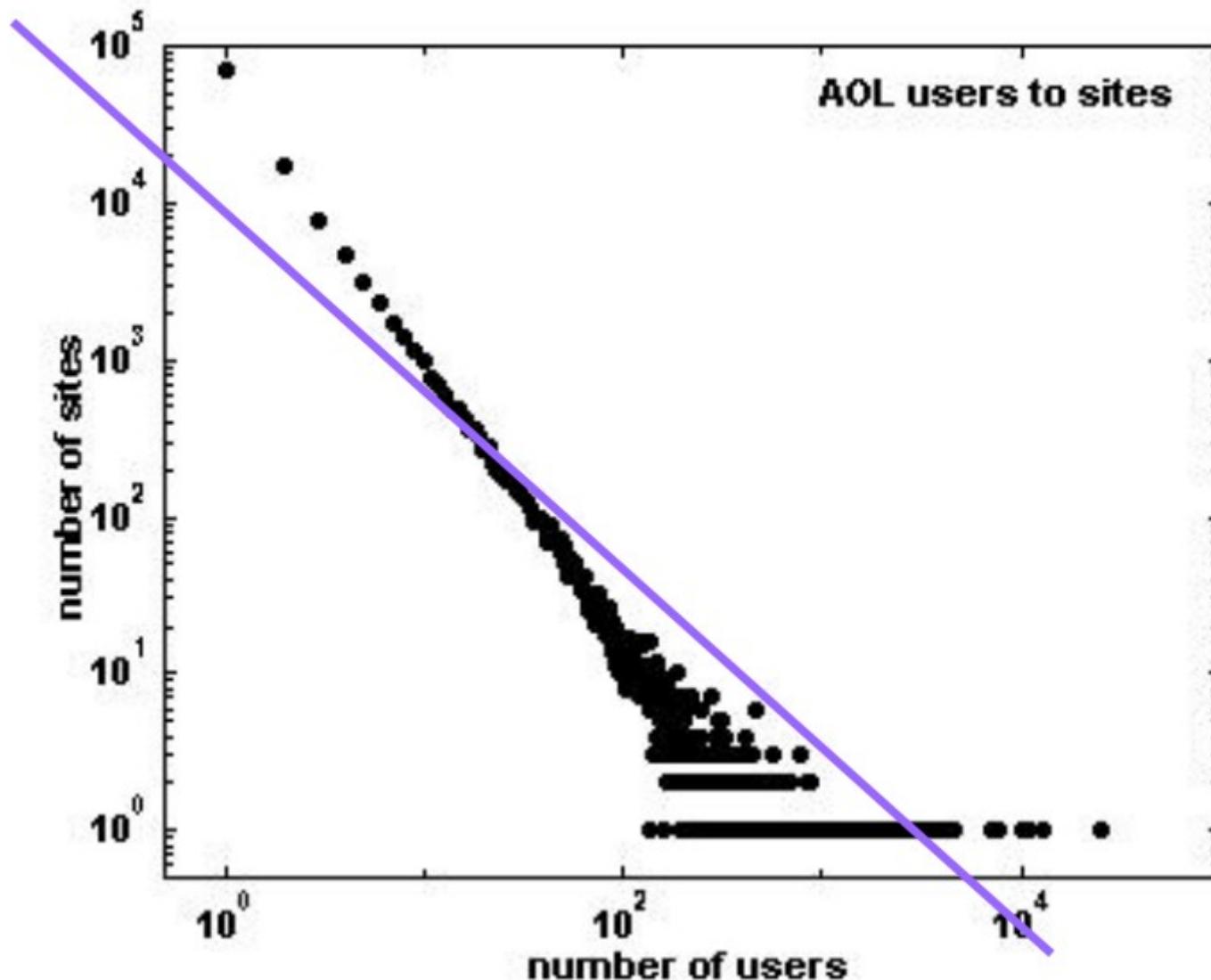
simple binning on a linear scale



simple binning on a log-log scale

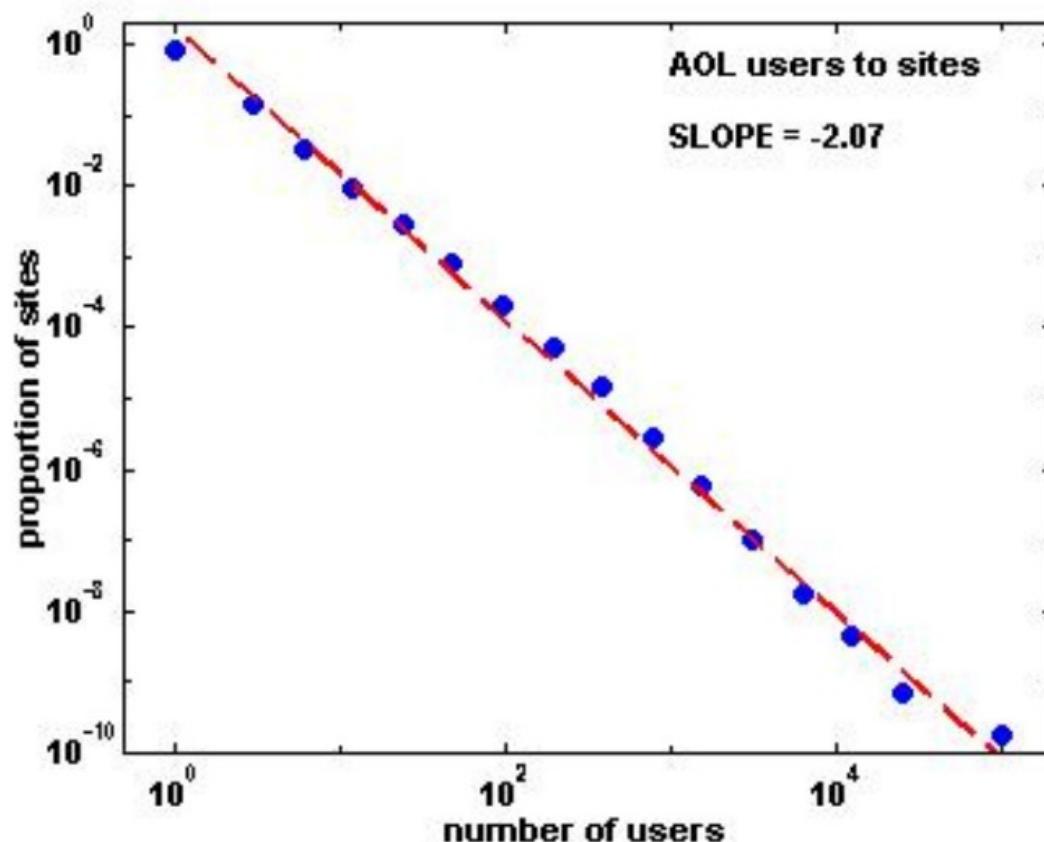
Example on a real data set

- Direct fit is too shallow: $\alpha = 1.17 \dots$



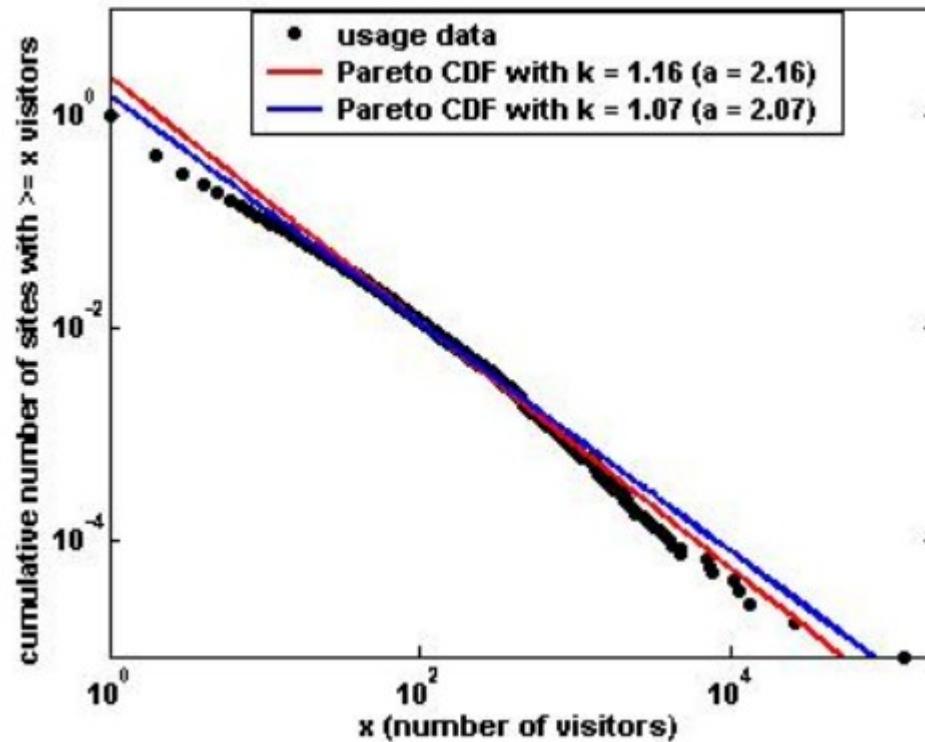
Example on a real data set

- **Binning logarithmically helps**
- Select exponentially wider bins
 - 1, 2, 4, 8, 16, 32,



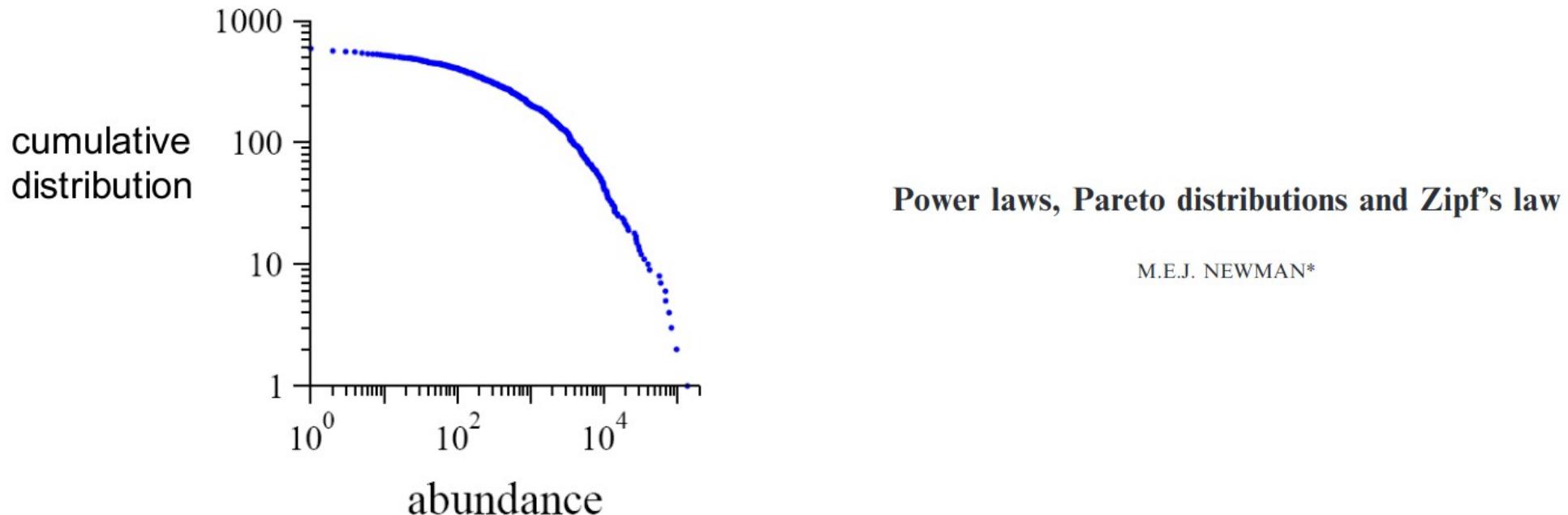
Example on a real data set

- Fitting the **cumulative distribution**
 - Shows perhaps 2 separate power-law regimes that were obscured by the exponential binning
 - Power-law tail may be closer to 2.4



Not everything is a power law!

- Number of **sightings of 591 bird species** in the North American Bird survey in 2003

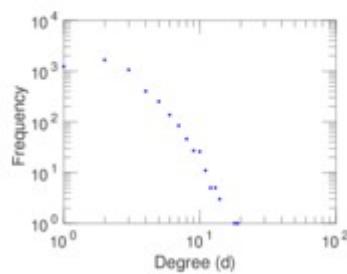


- another example:
 - **size of wildfires** (in acres)

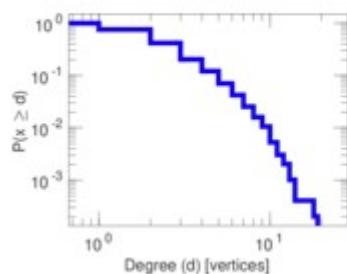
Not every network is power-law distributed

- Reciprocal, frequent email communication
- Power grid

Degree distribution



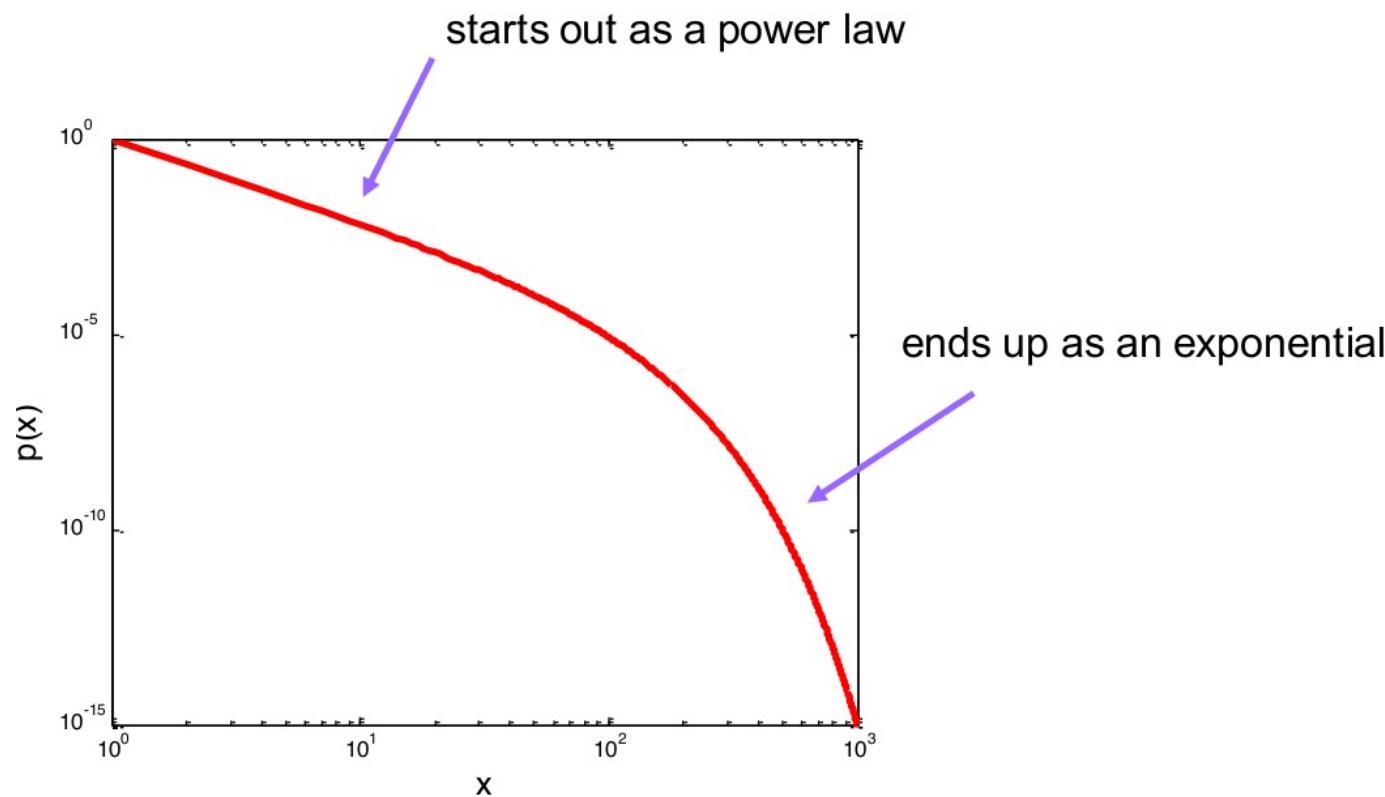
Cumulative degree distribution



- Company directors

Another common distribution

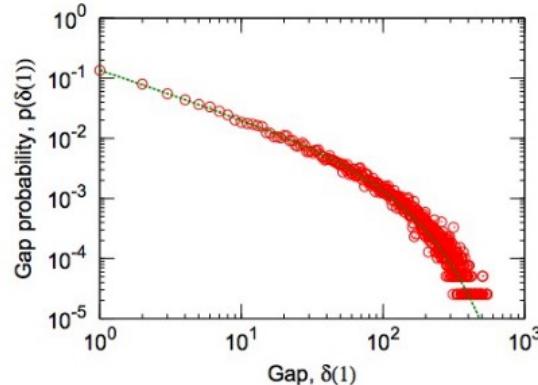
- Power-law with an exponential cutoff
 - $p(x) \sim x^{-\alpha} e^{-x/\kappa}$



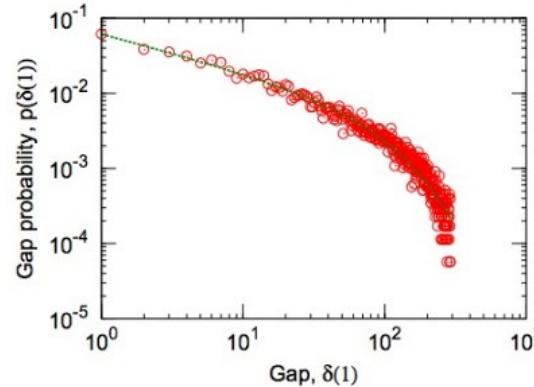
but could also be a lognormal or double exponential ...

Example of exponential cutoff

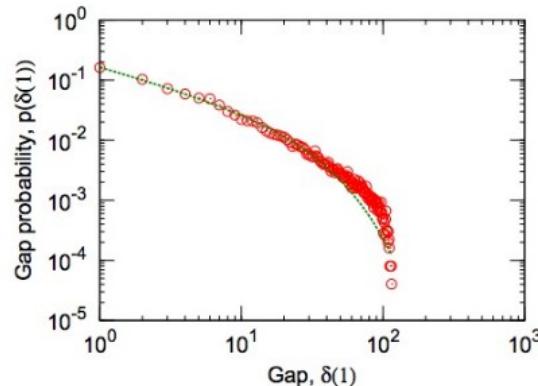
- Time between edge initiations



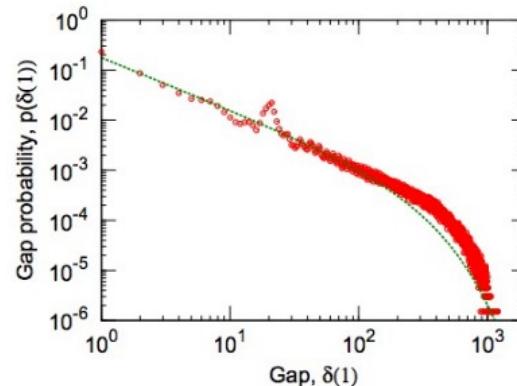
(a) FLICKR



(b) DELICIOUS



(c) ANSWERS



(d) LINKEDIN

Microscopic Evolution of Social Networks

Jure Leskovec* Lars Backstrom† Ravi Kumar‡ Andrew Tomkins‡

*Carnegie Mellon University †Cornell University ‡Yahoo Research

jure@cs.cmu.edu lars@cs.cornell.edu {ravikuma, atomkins}@yahoo-inc.com

Power-Laws: Wrap Up

- Power-laws are **cool** and **intriguing**

Power-law distributions in empirical data

Aaron Clauset,^{1,2} Cosma Rohilla Shalizi,³ and M. E. J. Newman⁴

¹Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA

²Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, USA

³Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, USA

⁴Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109, USA

Power-law distributions occur in many situations of scientific interest and have significant consequences for our understanding of natural and man-made phenomena. Unfortunately, the empirical detection and characterization of power laws is made difficult by the large fluctuations that occur in the tail of the distribution. In particular, standard methods such as least-squares fitting are known to produce systematically biased estimates of parameters for power-law distributions and should not be used in most circumstances. Here we describe statistical techniques for making accurate parameter estimates for power-law data, based on maximum likelihood methods and the Kolmogorov-Smirnov statistic. We also show how to tell whether the data follow a power-law distribution at all, defining quantitative measures that indicate when the power law is a reasonable fit to the data and when it is not. We demonstrate these methods by applying them to twenty-four real-world data sets from a range of different disciplines. Each of the data sets has been conjectured previously to follow a power-law distribution. In some cases we find these conjectures to be consistent with the data while in others the power law is ruled out.

- But make sure your data is actually power-law before boasting!

ARTICLE

<https://doi.org/10.1038/s41467-019-08746-5>

OPEN

Scale-free networks are rare

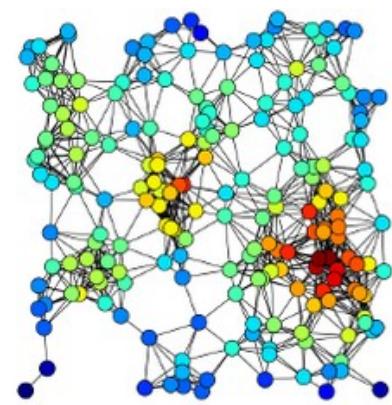
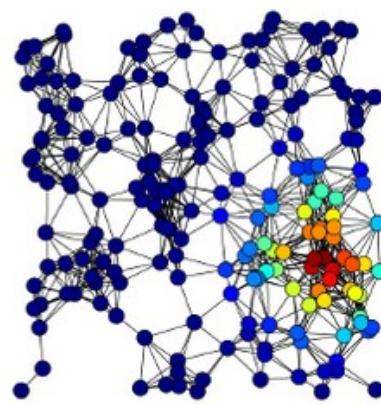
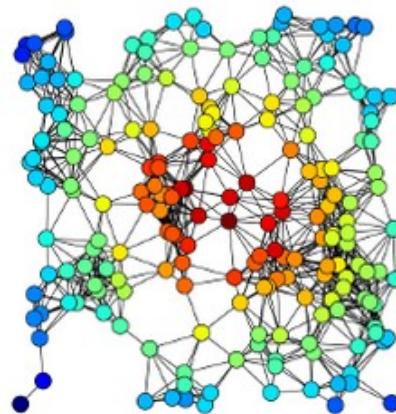
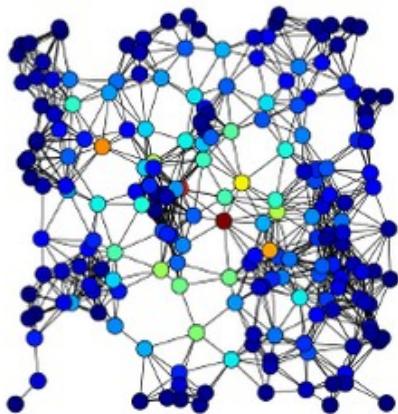
Anna D. Broido¹ & Aaron Clauset^{2,3,4}

Real-world networks are often claimed to be scale free, meaning that the fraction of nodes with degree k follows a power law $k^{-\alpha}$, a pattern with broad implications for the structure and dynamics of complex systems. However, the universality of scale-free networks remains controversial. Here, we organize different definitions of scale-free networks and construct a

Node Centrality



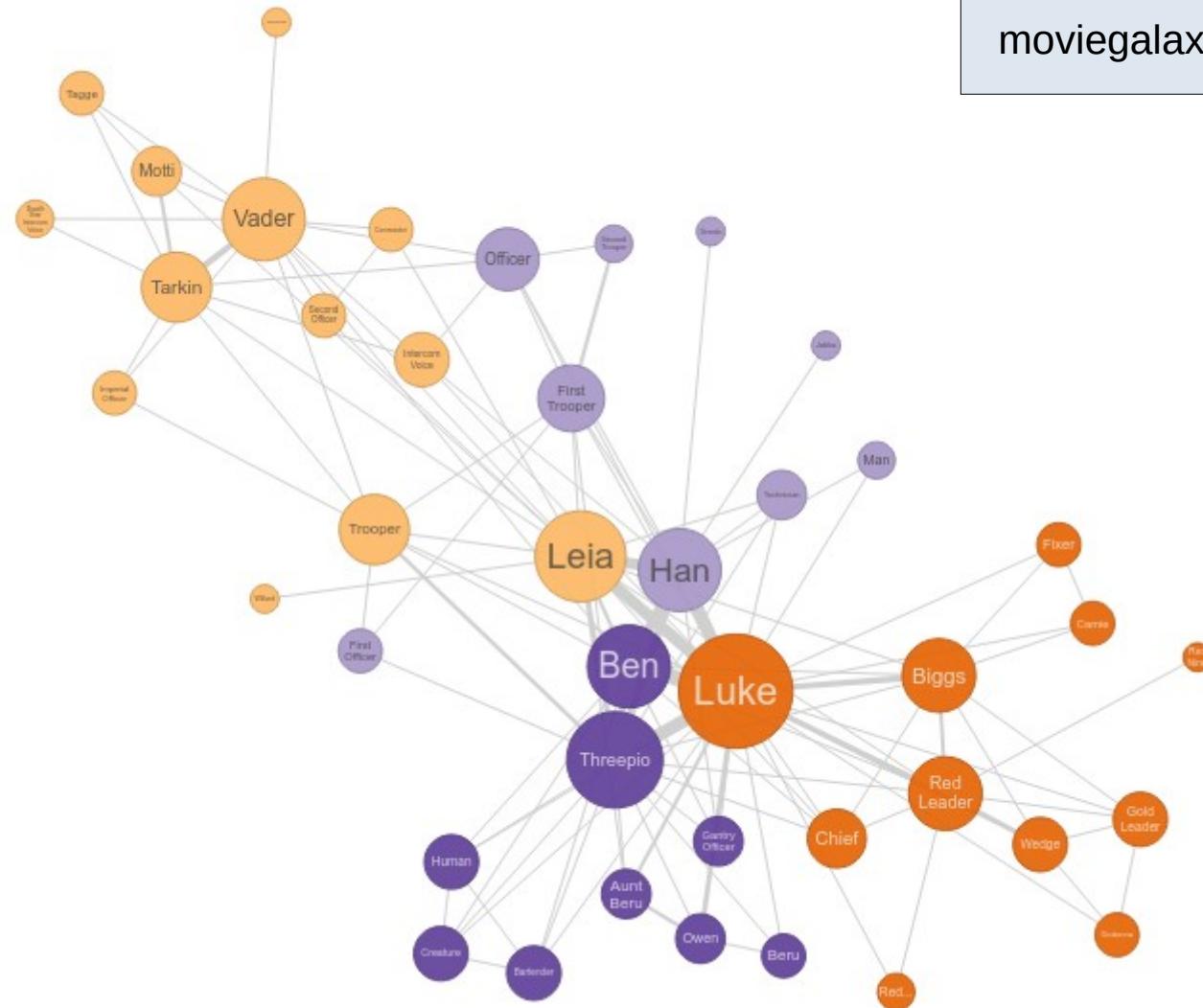
Pedro Ribeiro
(DCC/FCUP & CRACS/INESC-TEC)



(Heavily based on slides from Jure Leskovec and Lada Adamic @ Stanford University)

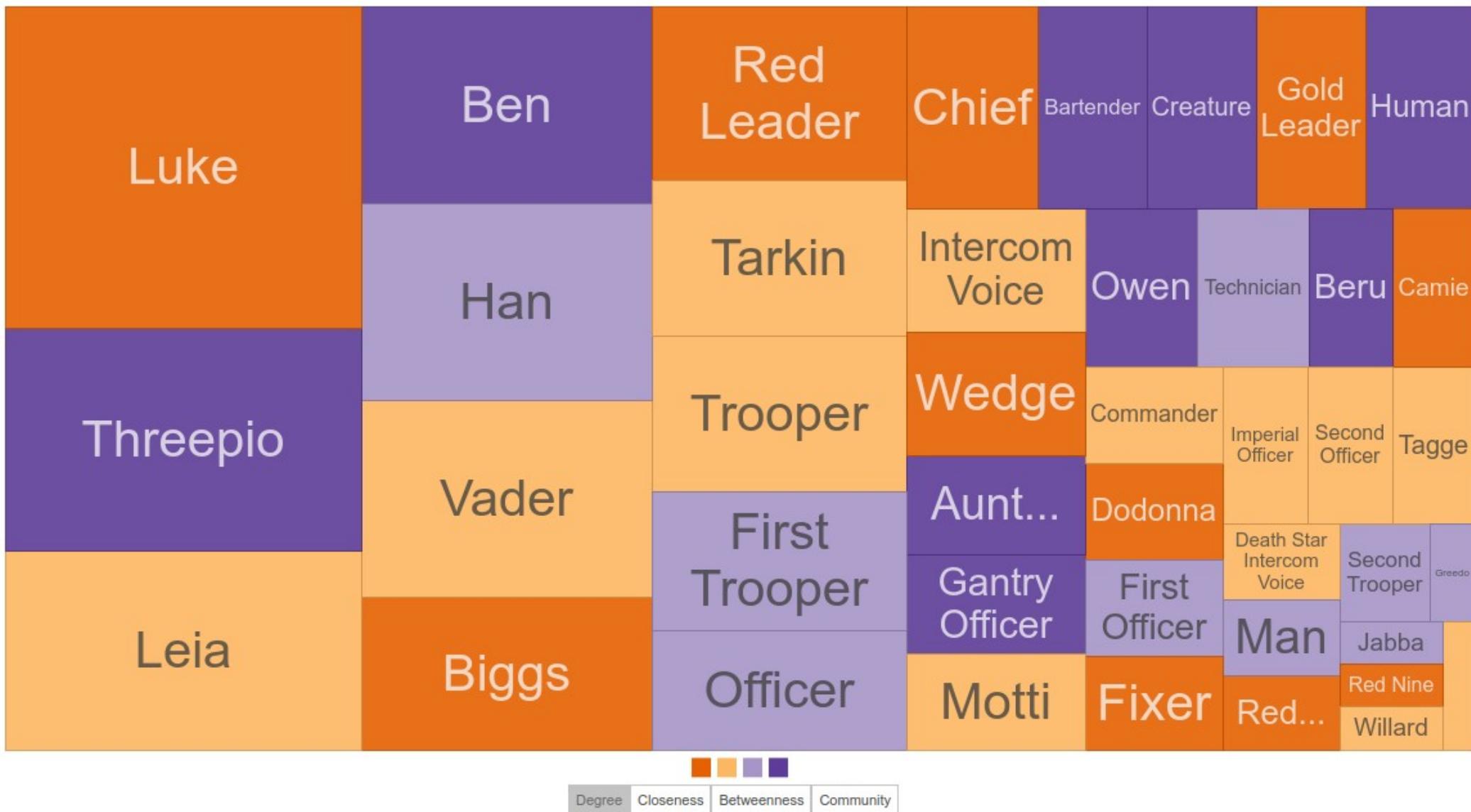
Star Wars IV Network

moviegalaxies.com



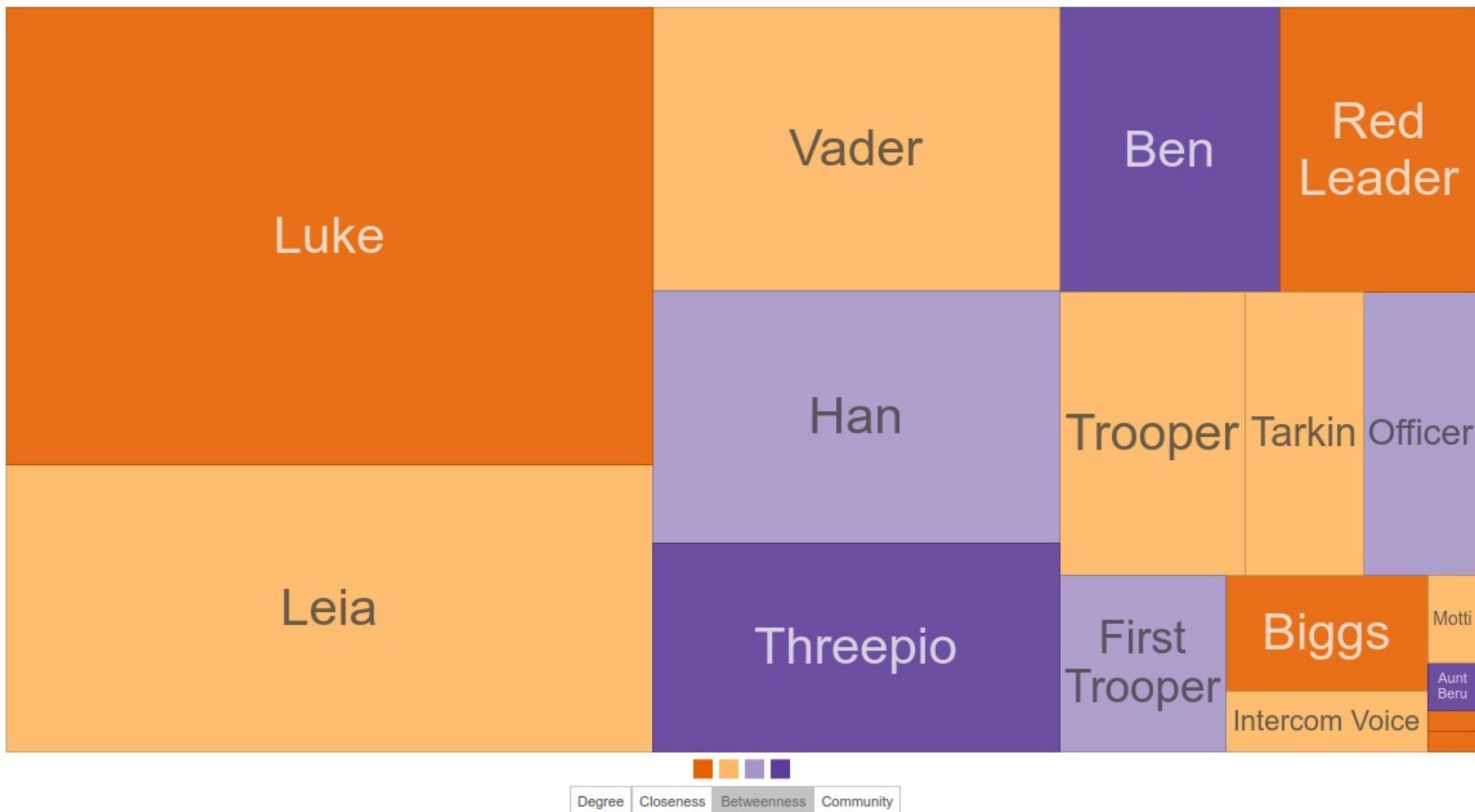
Are all nodes “equal”? How to measure their importance?

Star Wars IV Network



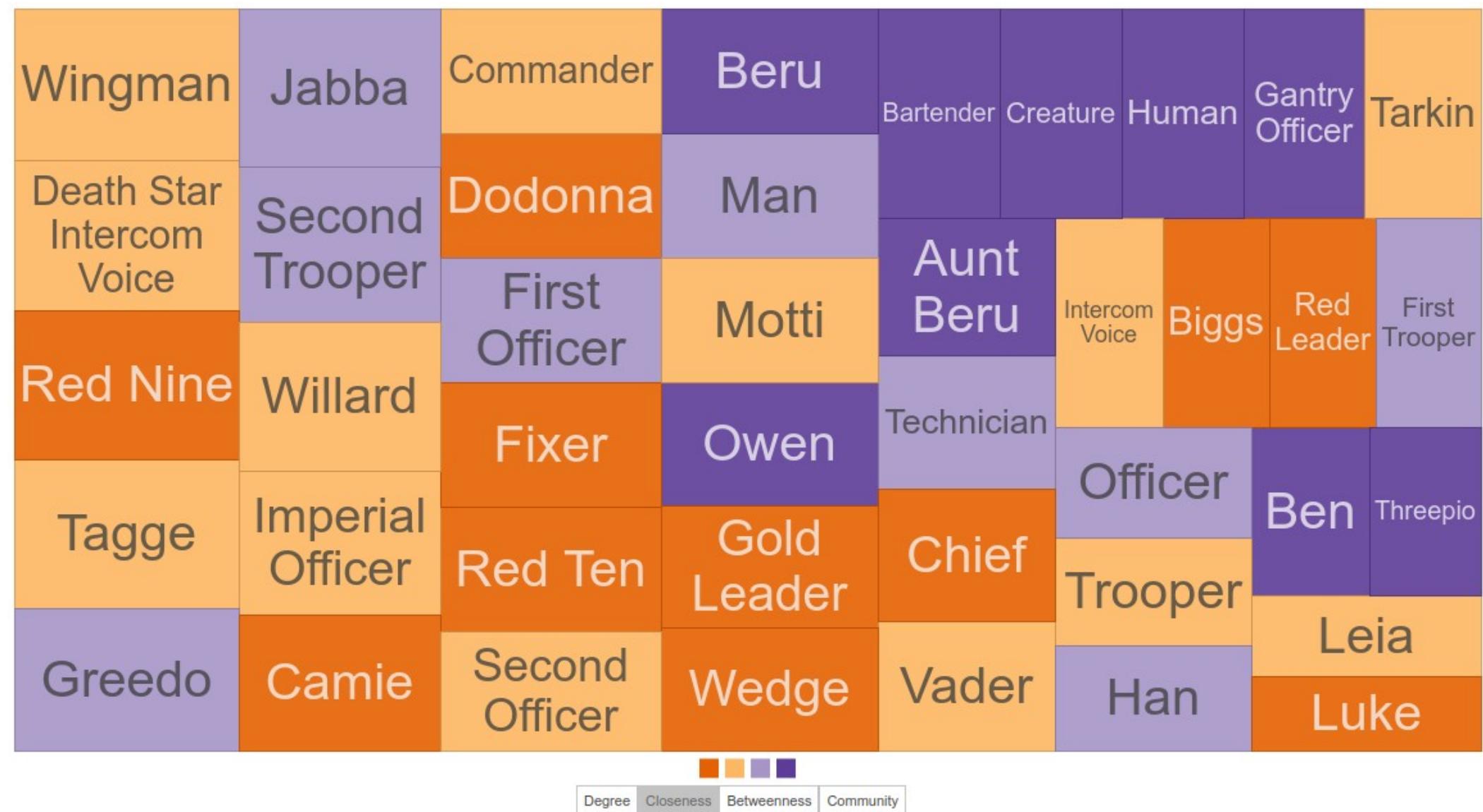
Size proportional to degree: is this the only way?

Star Wars IV Network



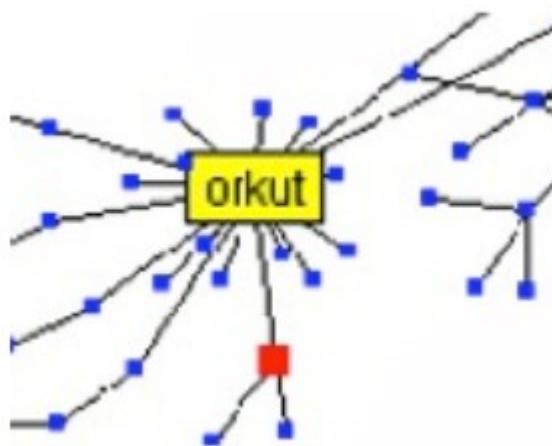
Size proportional to betweenness

Star Wars IV Network



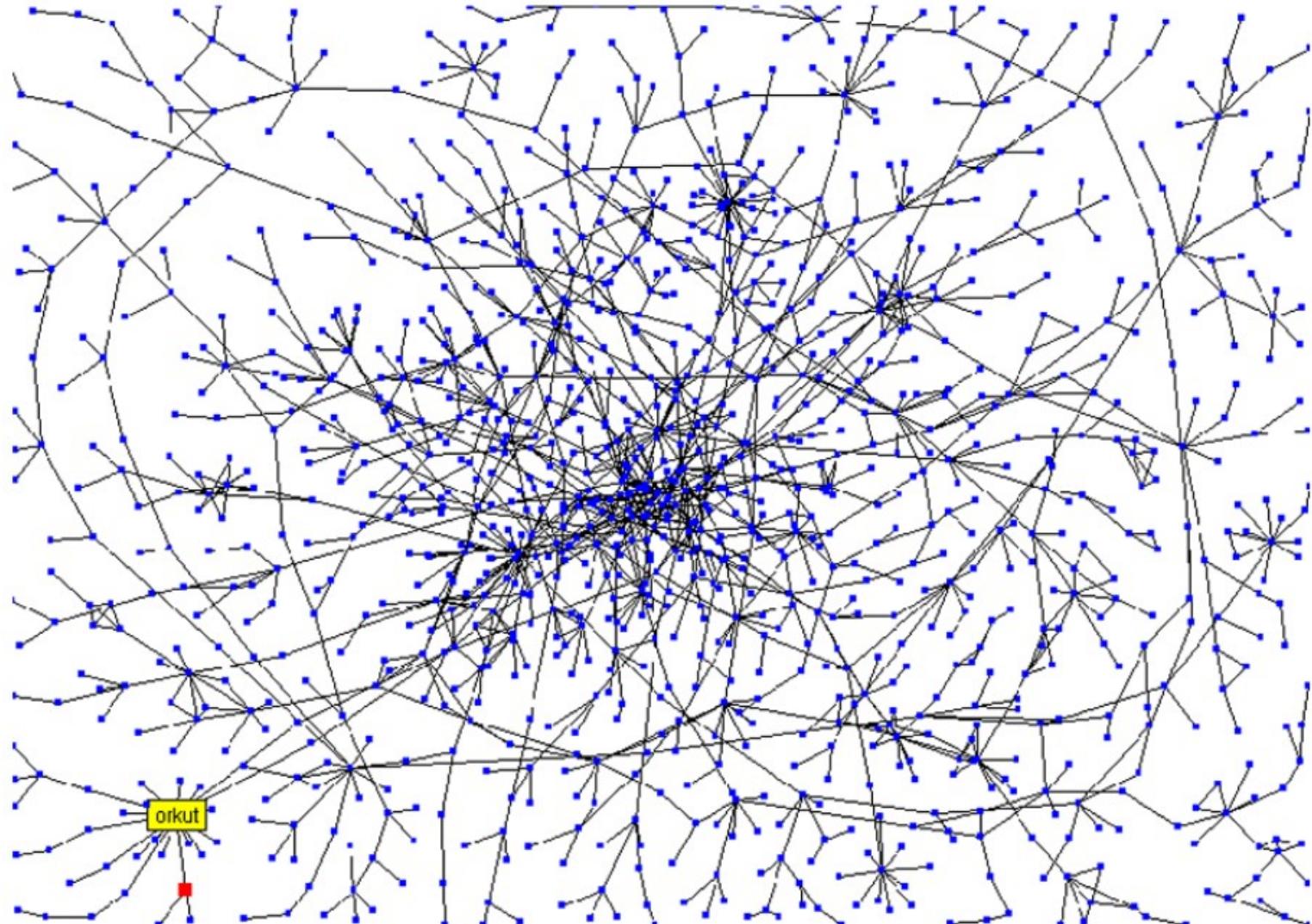
Size proportional to closeness

Why degree is not enough



Why degree is not enough

Stanford Social Web (ca. 1999)



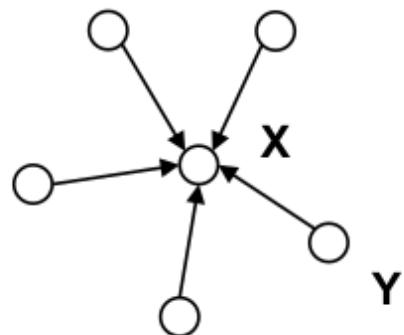
network of personal homepages at Stanford

Pedro Ribeiro - Node Centrality

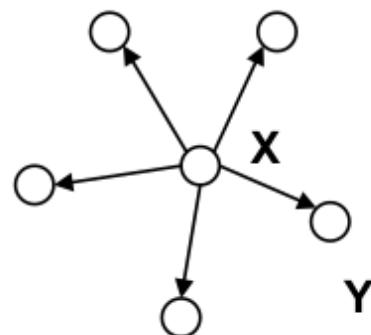
Different notions of centrality

- **Node Centrality** measures “importance”

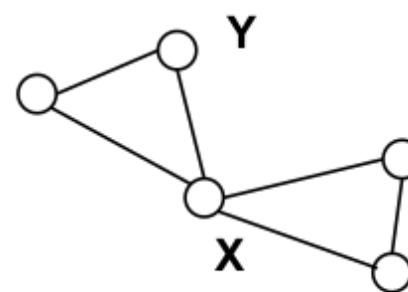
In each of the following networks, X has higher centrality than Y according to a particular measure



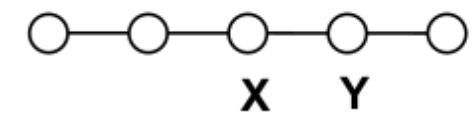
indegree



outdegree



betweenness



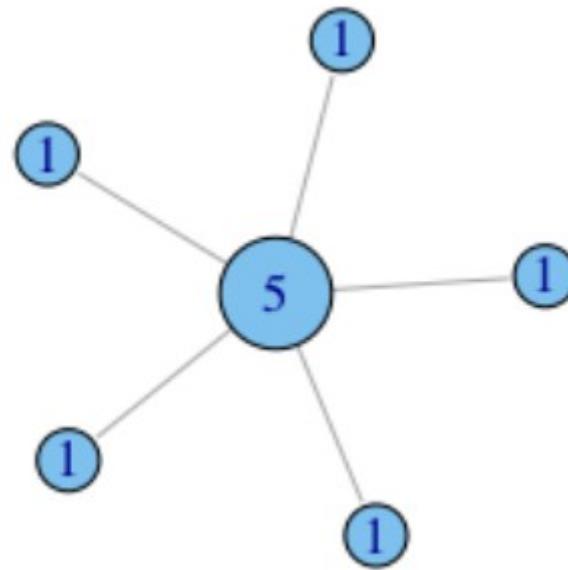
closeness

Node Degree

- Let's put some **numbers** to it

Undirected degree:

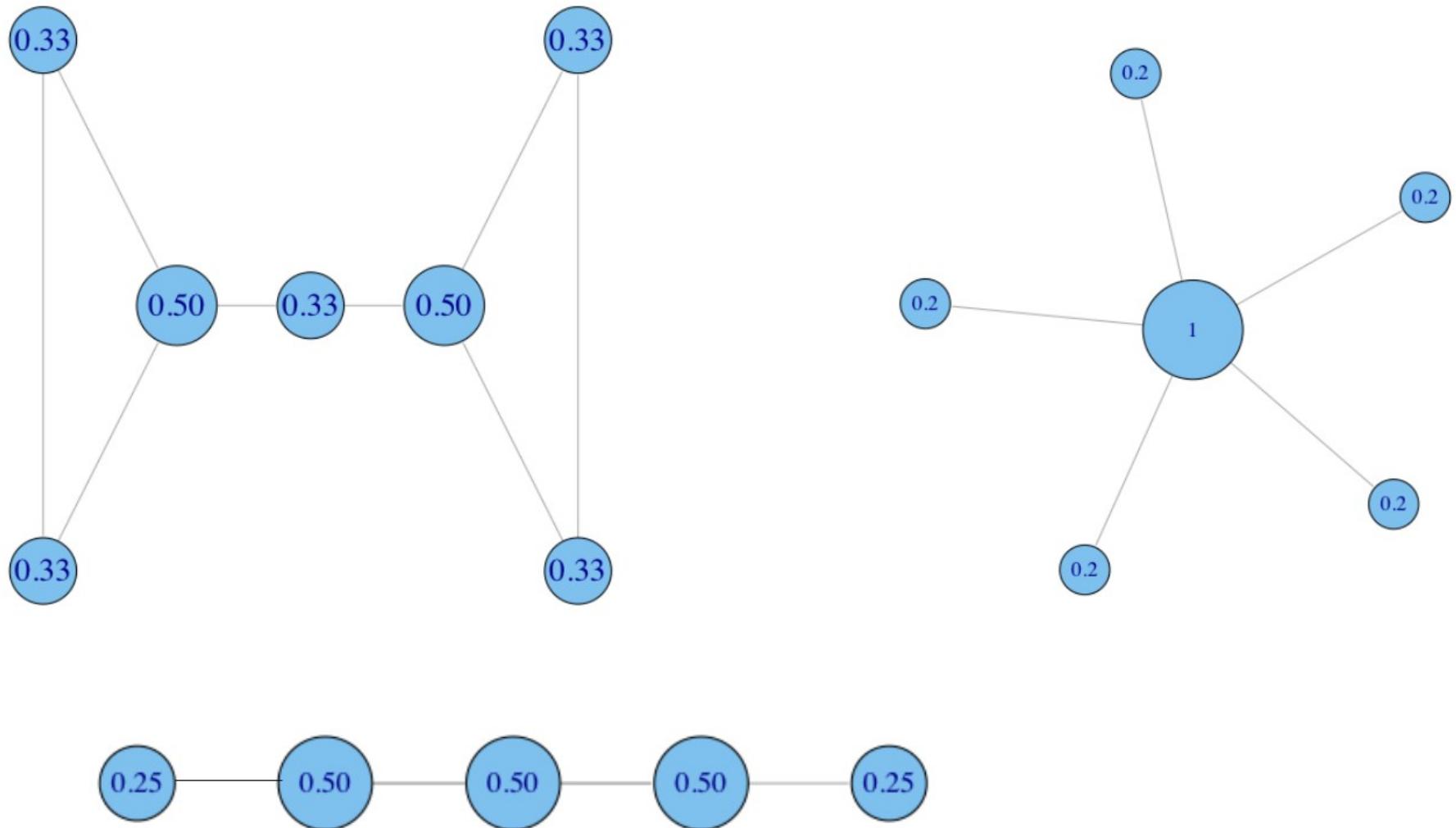
e.g. nodes with more friends are more central.



Assumption: the connections that your friend has don't matter, it is what they can do directly that does (e.g. go have a beer with you, help you build a deck...)

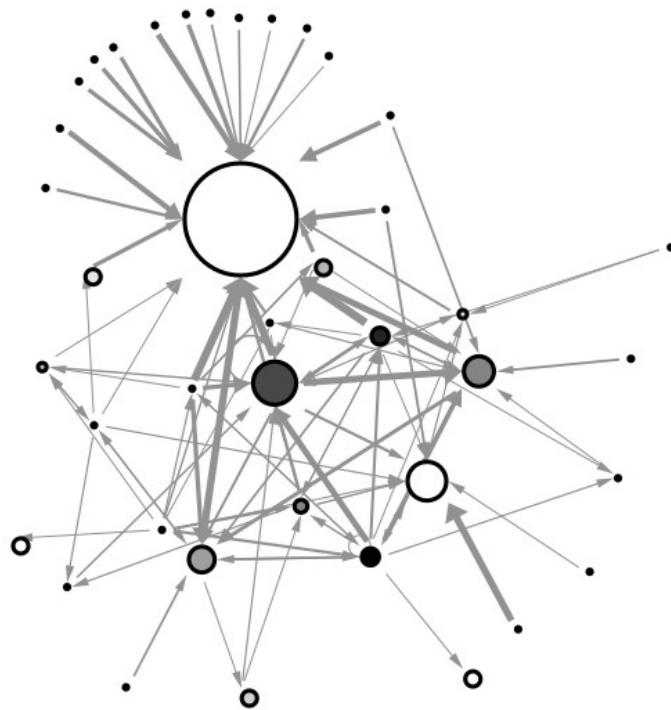
Node Degree

- **Normalization:**
divide degree by the max. possible, i.e. $(N-1)$

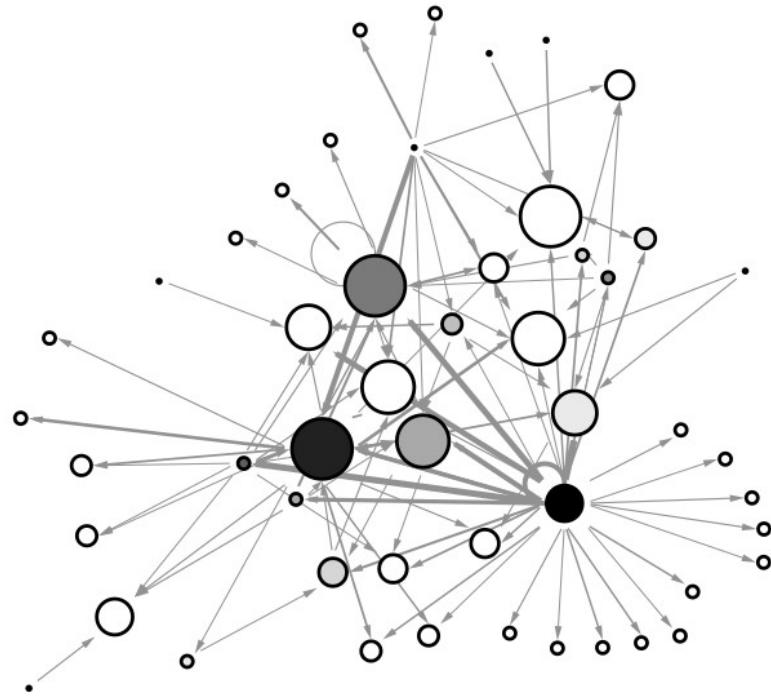


Node Degree

example financial trading networks



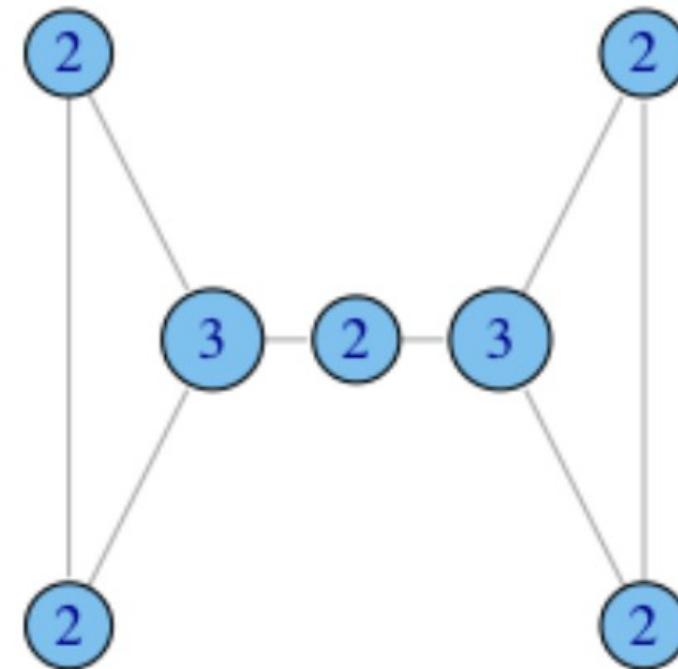
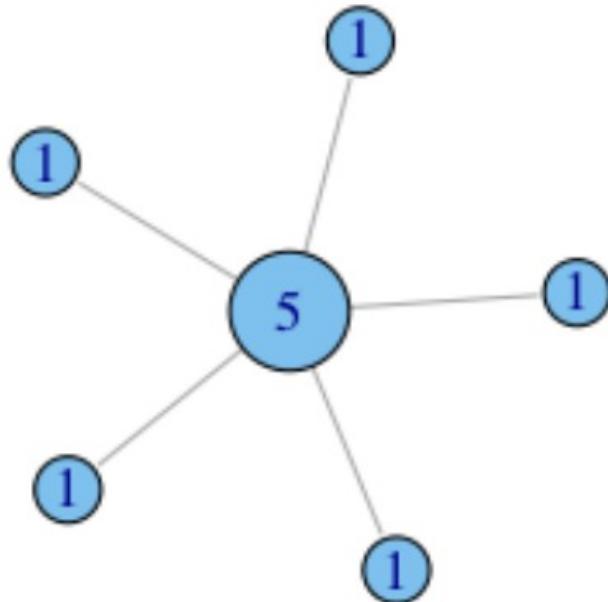
high in-centralization:
one node buying from
many others



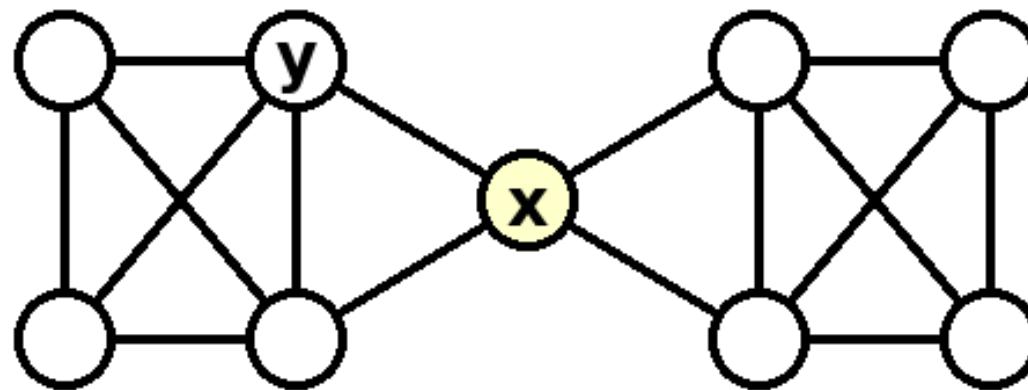
low in-centralization:
buying is more evenly
distributed

What does degree not capture?

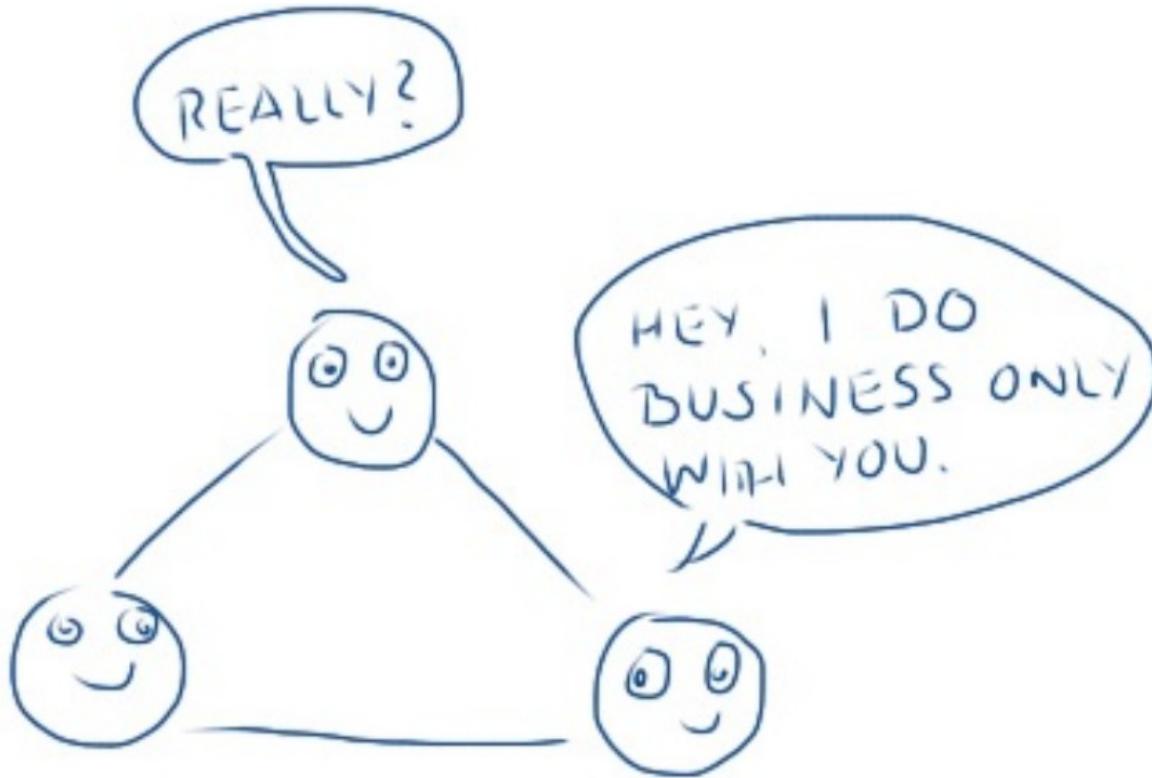
- In what ways does degree fail to capture centrality in the following graphs?



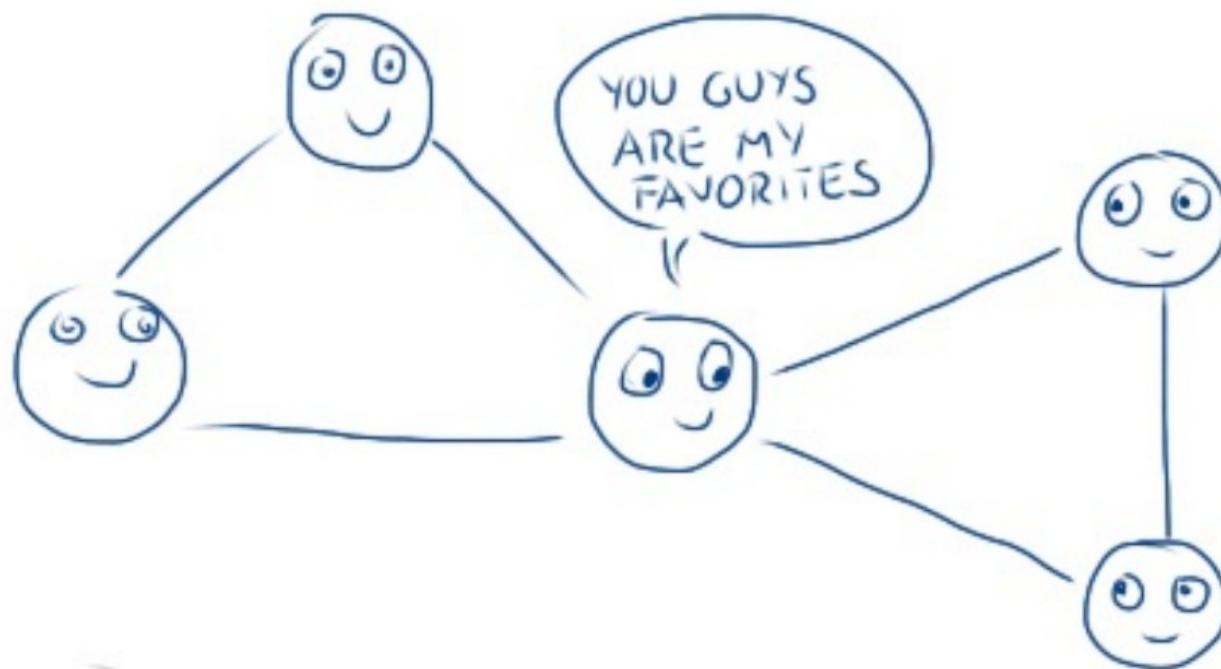
Brokerage not captured by degree



Brokerage: Concept



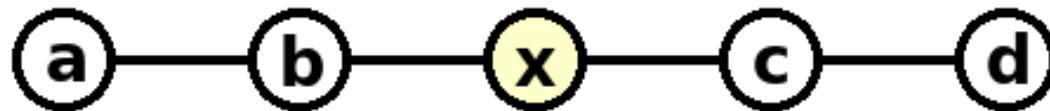
Brokerage: Concept



Capturing Brokerage

- **Betweenness Centrality:**

intuition: how many **pairs of individuals** would have to go through you in order to reach one another in the **minimum number of hops**?



Betweenness: Definition

$$C_B(i) = \sum_{j < k} \frac{g_{jk}(i)}{g_{jk}}$$

Where:

g_{jk} = the number of **shortest paths** connecting nodes j and k

$g_{jk}(i)$ = the number that node i is on.

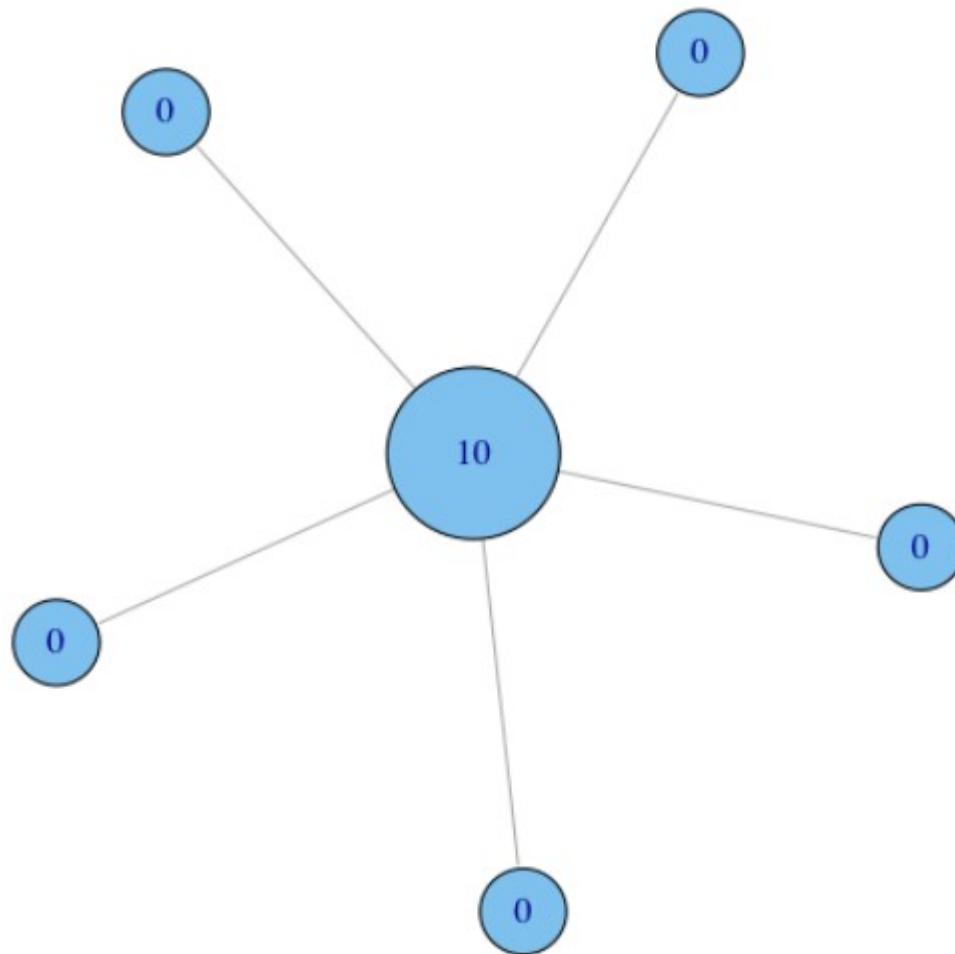
Usually normalized by:

$$C'_B(i) = \frac{C_B(i)}{(n-1)(n-2)/2}$$

number of pairs of vertices
excluding the vertex itself

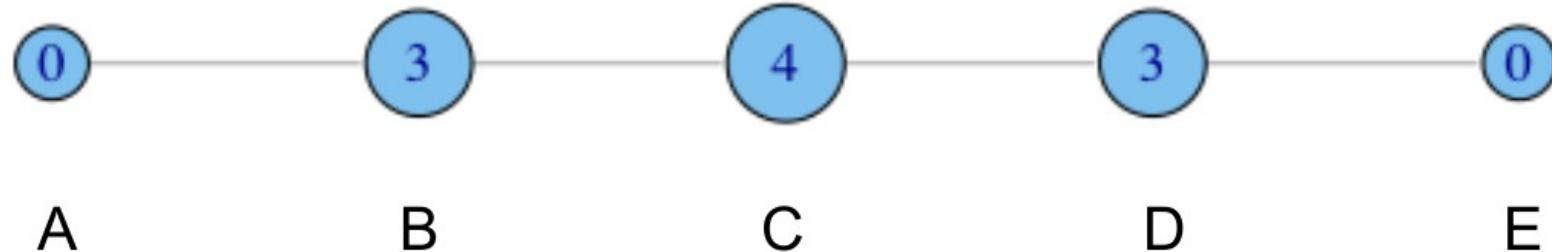
Betweenness: Toy Networks

- Non-normalized version:



Betweenness: Toy Networks

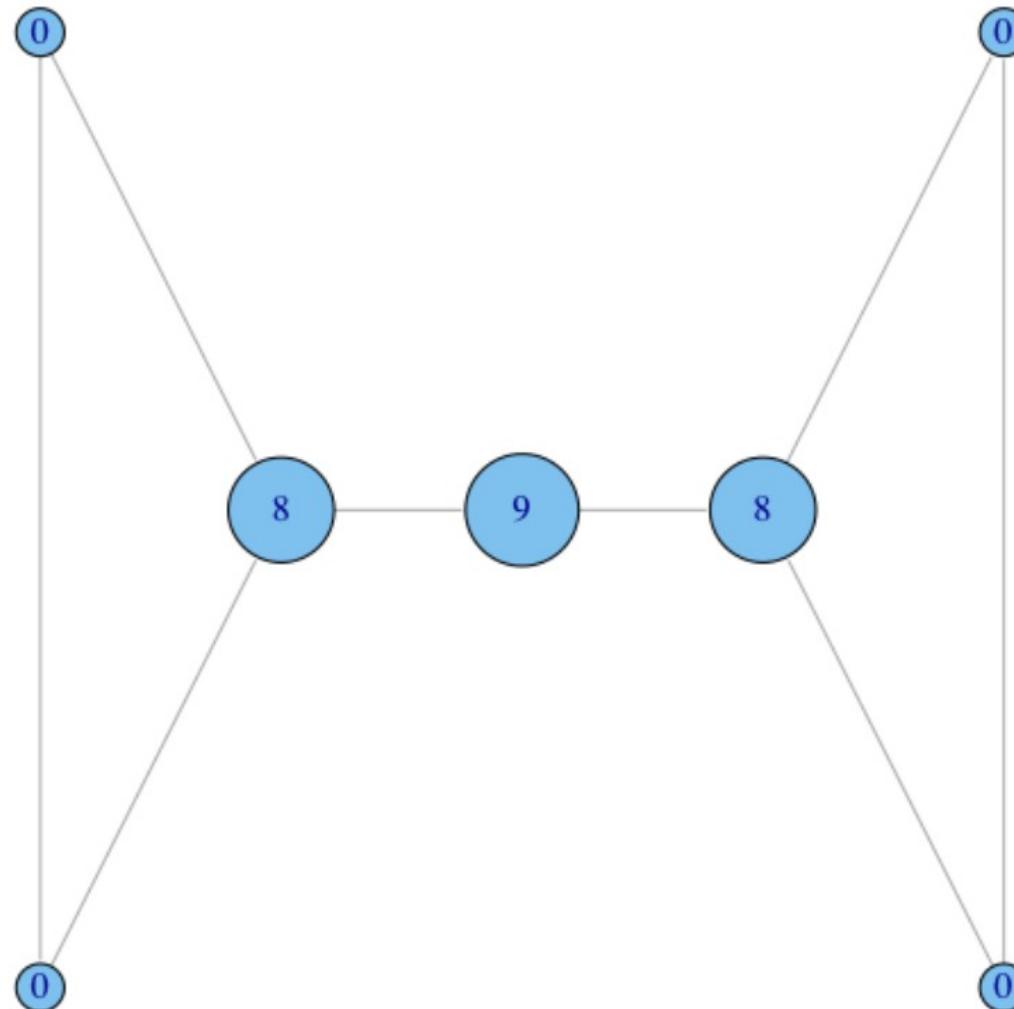
- Non-normalized version:



- A lies between no two other vertices
- B lies between A and 3 other vertices: C, D, and E
- C lies between 4 pairs of vertices: (A,D),(A,E),(B,D),(B,E)
 - note that there are no alternate paths for these pairs to take, so C gets full credit

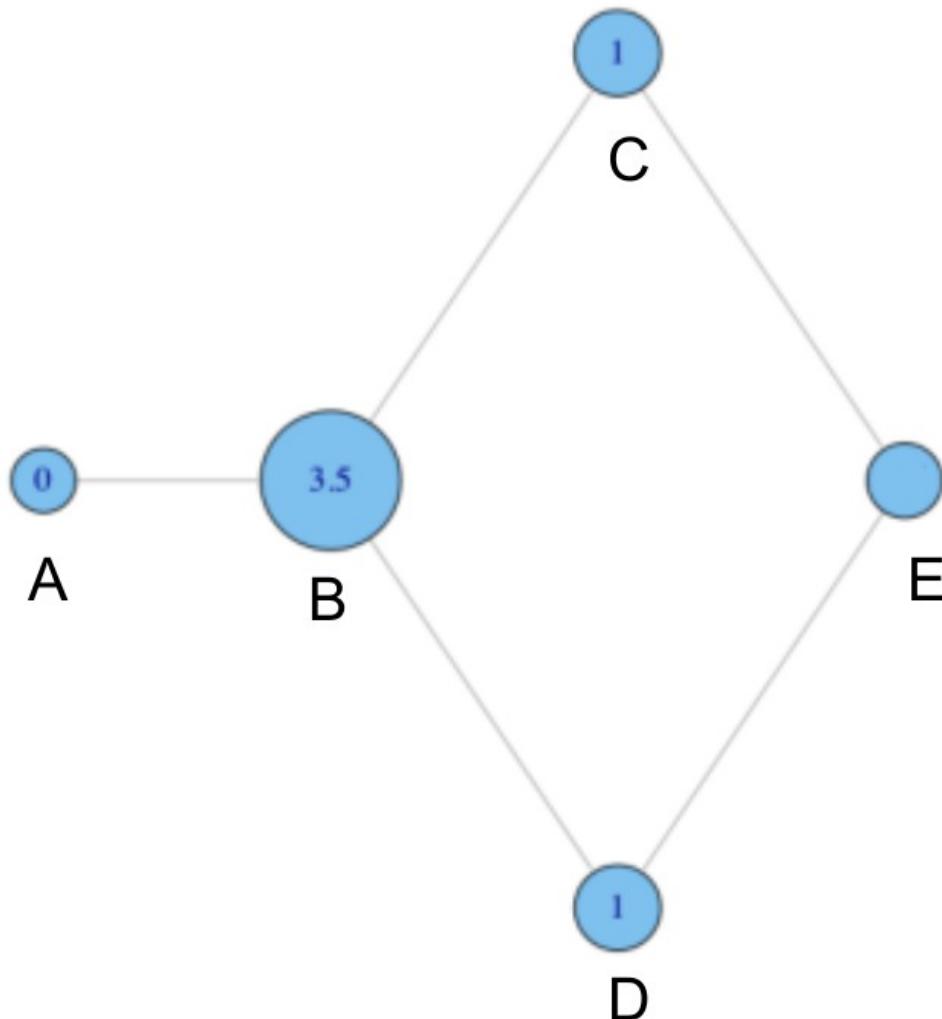
Betweenness: Toy Networks

- Non-normalized version:



Betweenness: Toy Networks

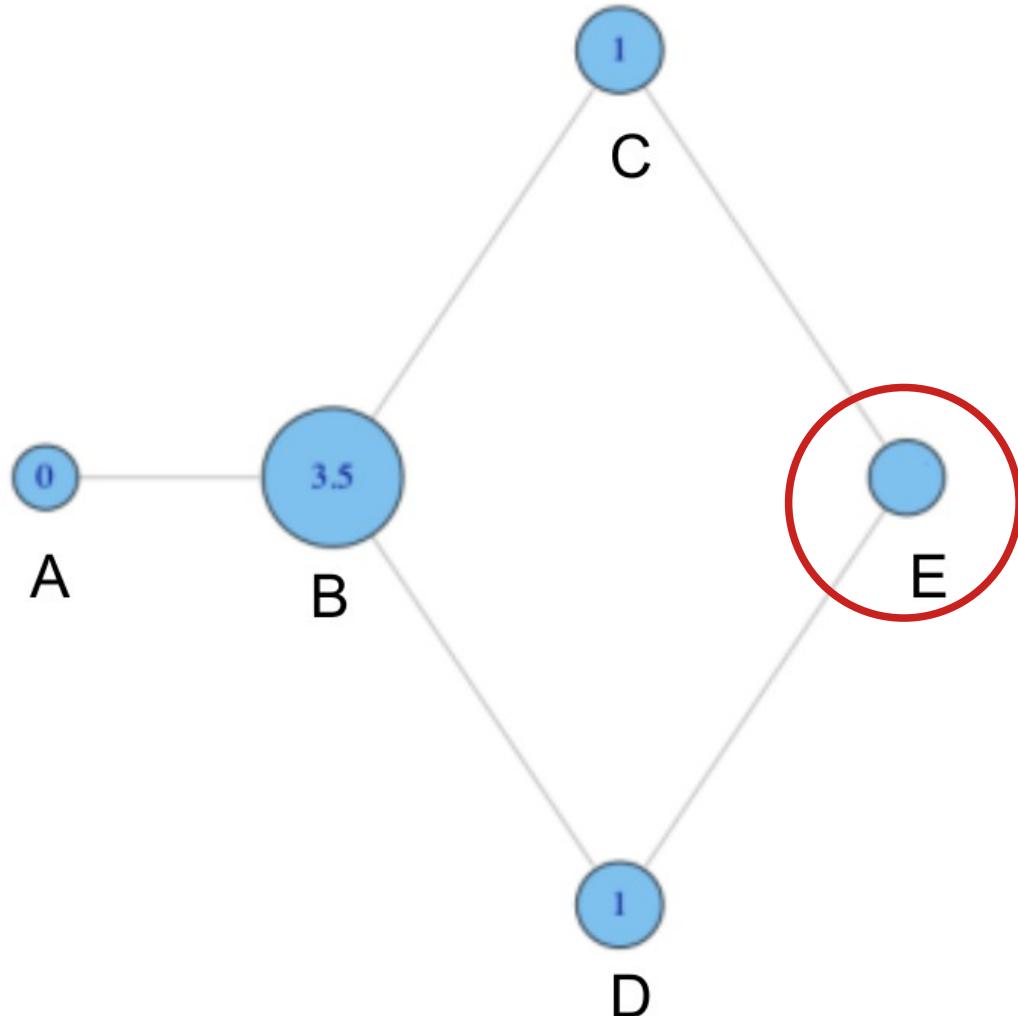
- Non-normalized version:



- why do C and D each have betweenness 1?
- They are both on shortest paths for pairs (A,E), and (B,E), and so must share credit:
 - $1/2+1/2 = 1$

Betweenness: Toy Networks

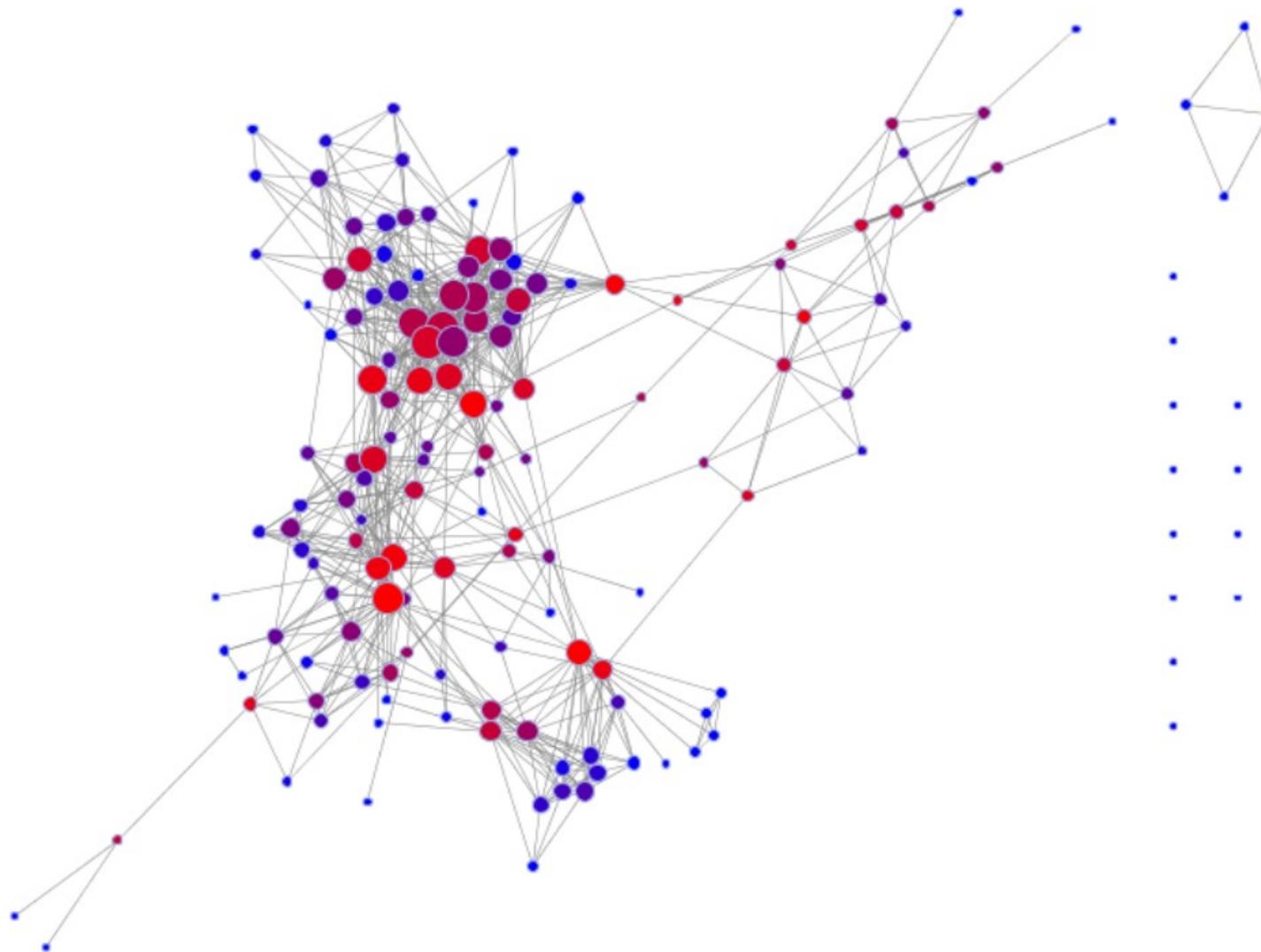
- Non-normalized version:



What is the betweenness
of node E?

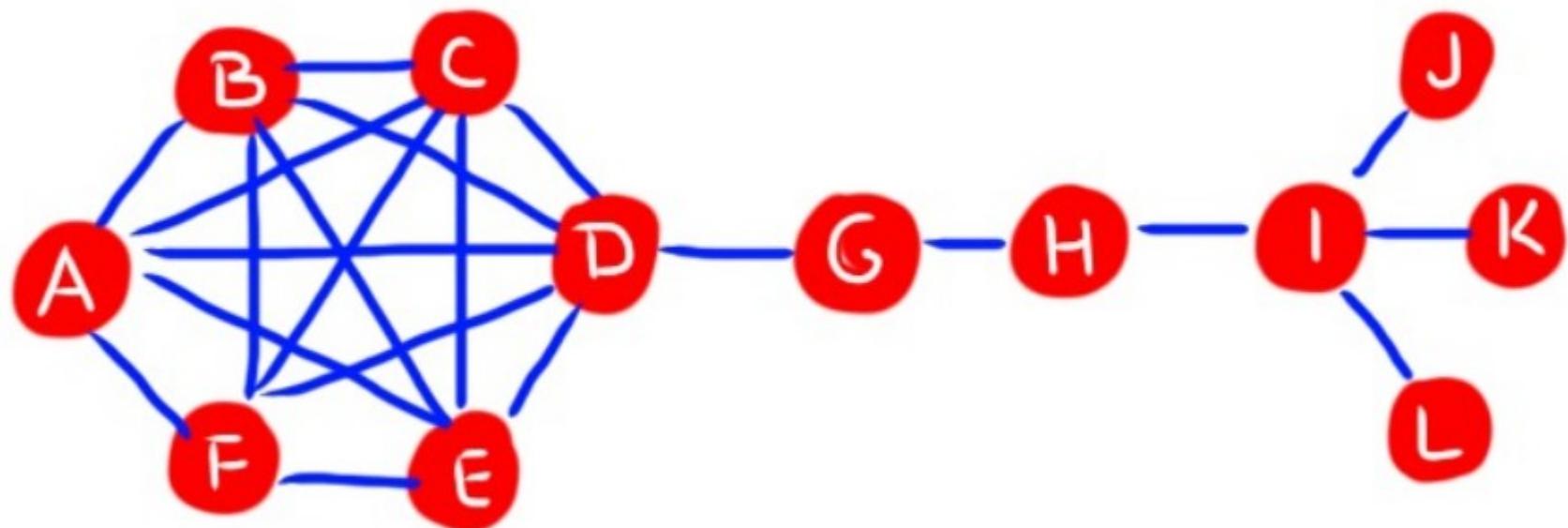
Betweenness: Real Example

- Social Network (facebook)
nodes are sized by degree, and colored by betweenness



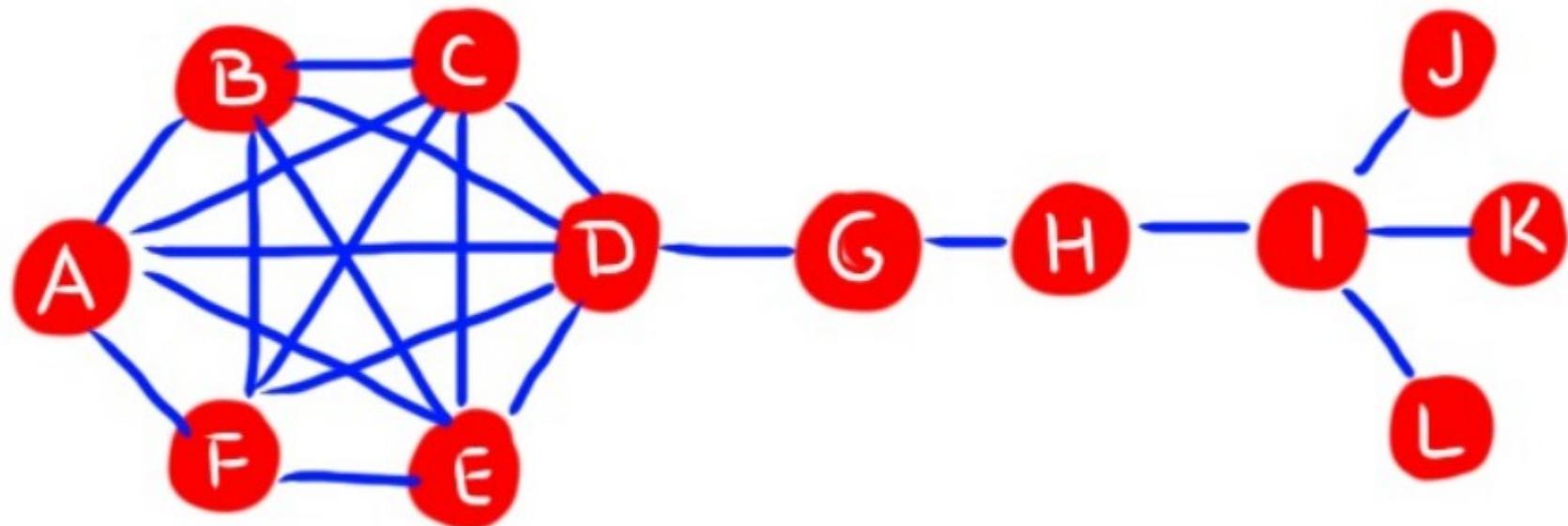
Betweenness: Question

- Find a node that has **high betweenness** but **low degree**



Betweenness: Question

- Find a node that has **low betweenness** but **high degree**

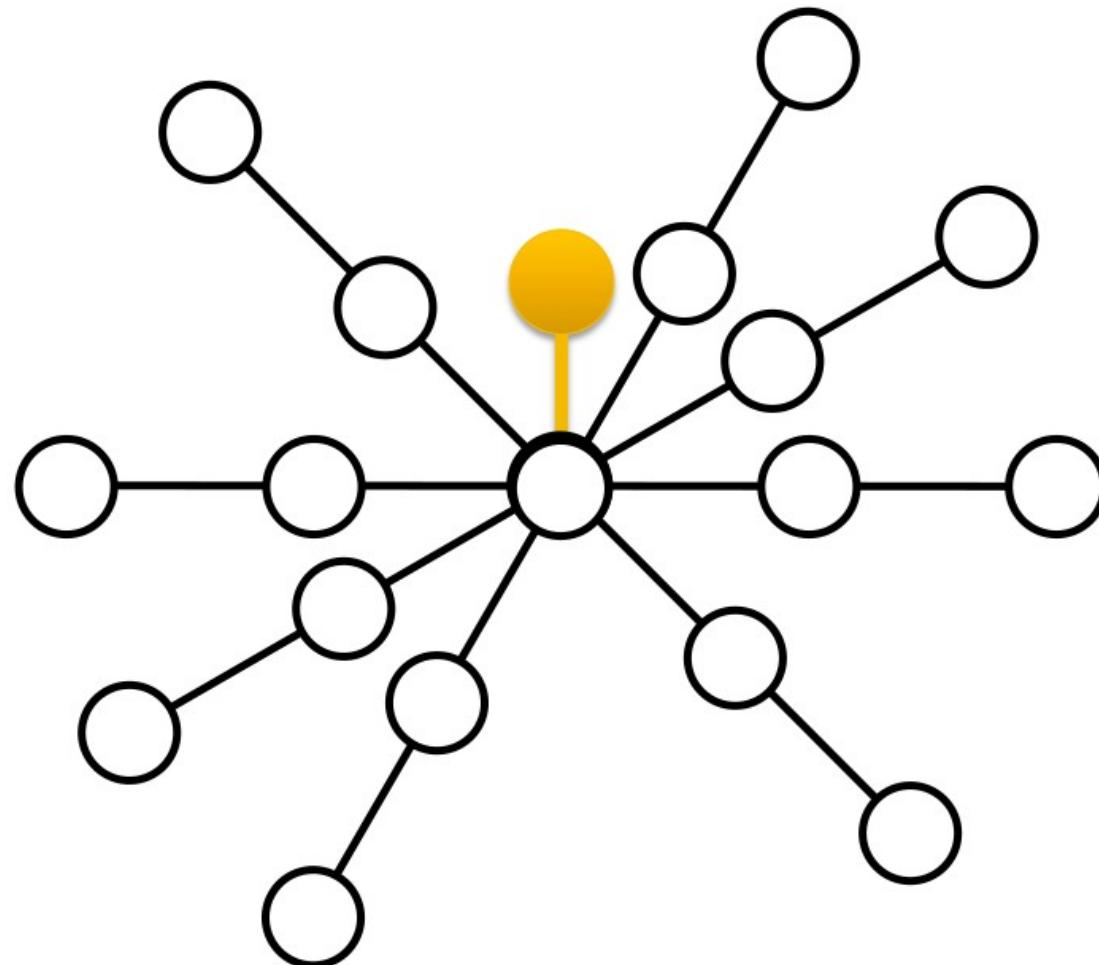


Closeness Centrality

- What if it's not so important to have many direct friends?
- Or be “between” others
- But one still wants to be in the “**middle**” of things, **not too far from the center**

Closeness Centrality

- Need not be in brokerage position



Closeness: Definition

- **Closeness** is based on the **length of the average shortest path** between a node and all other nodes in the network

Closeness Centrality:

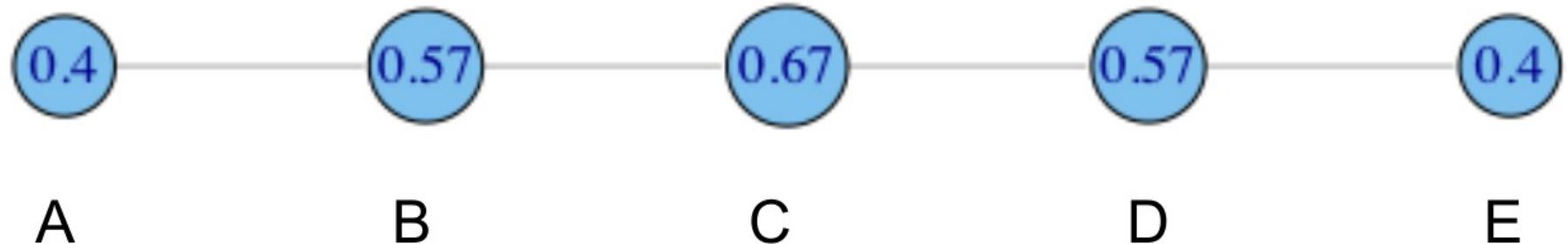
$$C_C(i) = \frac{1}{N} \sum_{j=1}^N d(i, j)$$

Normalized Closeness Centrality:

$$C'_C(i) = C_C(i) \times (n - 1)$$

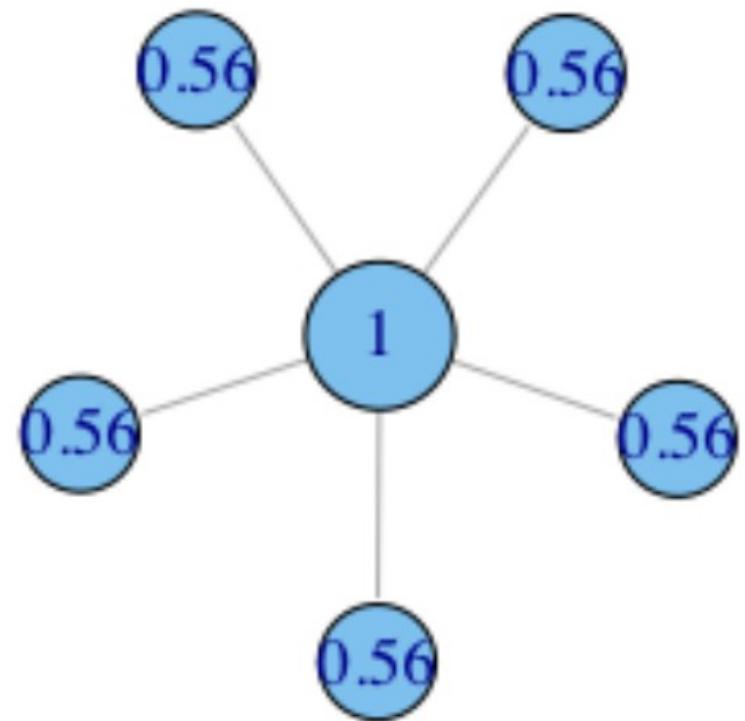
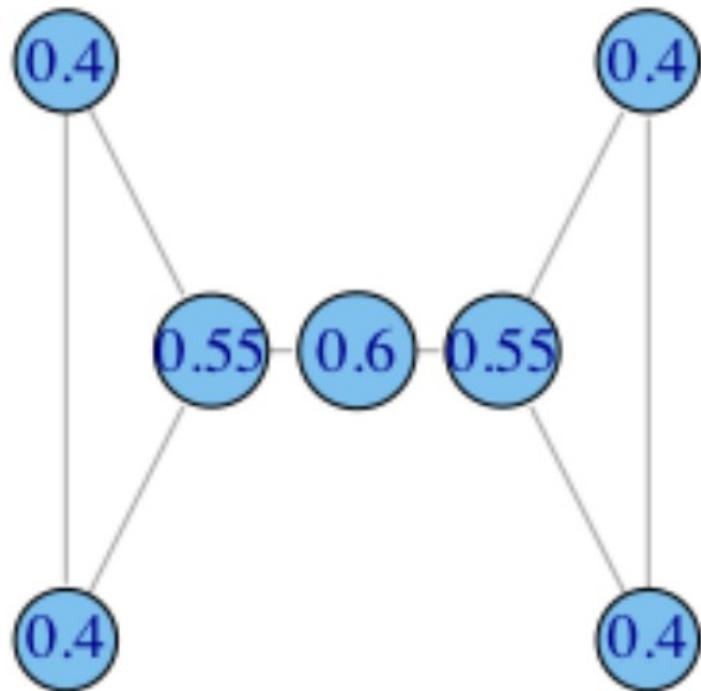
When graphs are big, the -1 can be discarded and we multiply by n

Closeness: Toy Networks



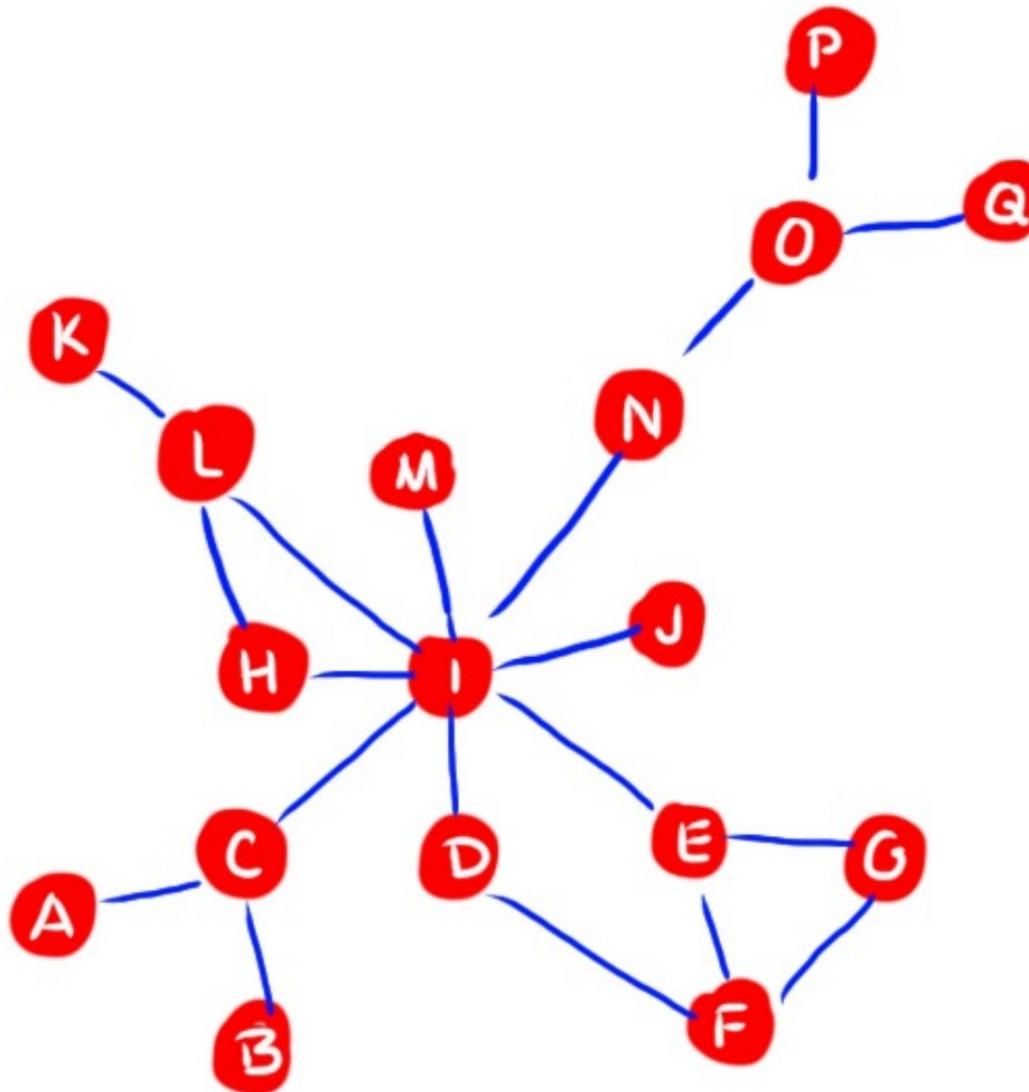
$$C_c(A) = \left[\frac{\sum_{j=1}^N d(A, j)}{N - 1} \right]^{-1} = \left[\frac{1 + 2 + 3 + 4}{4} \right]^{-1} = \left[\frac{10}{4} \right]^{-1} = 0.4$$

Closeness: Toy Networks



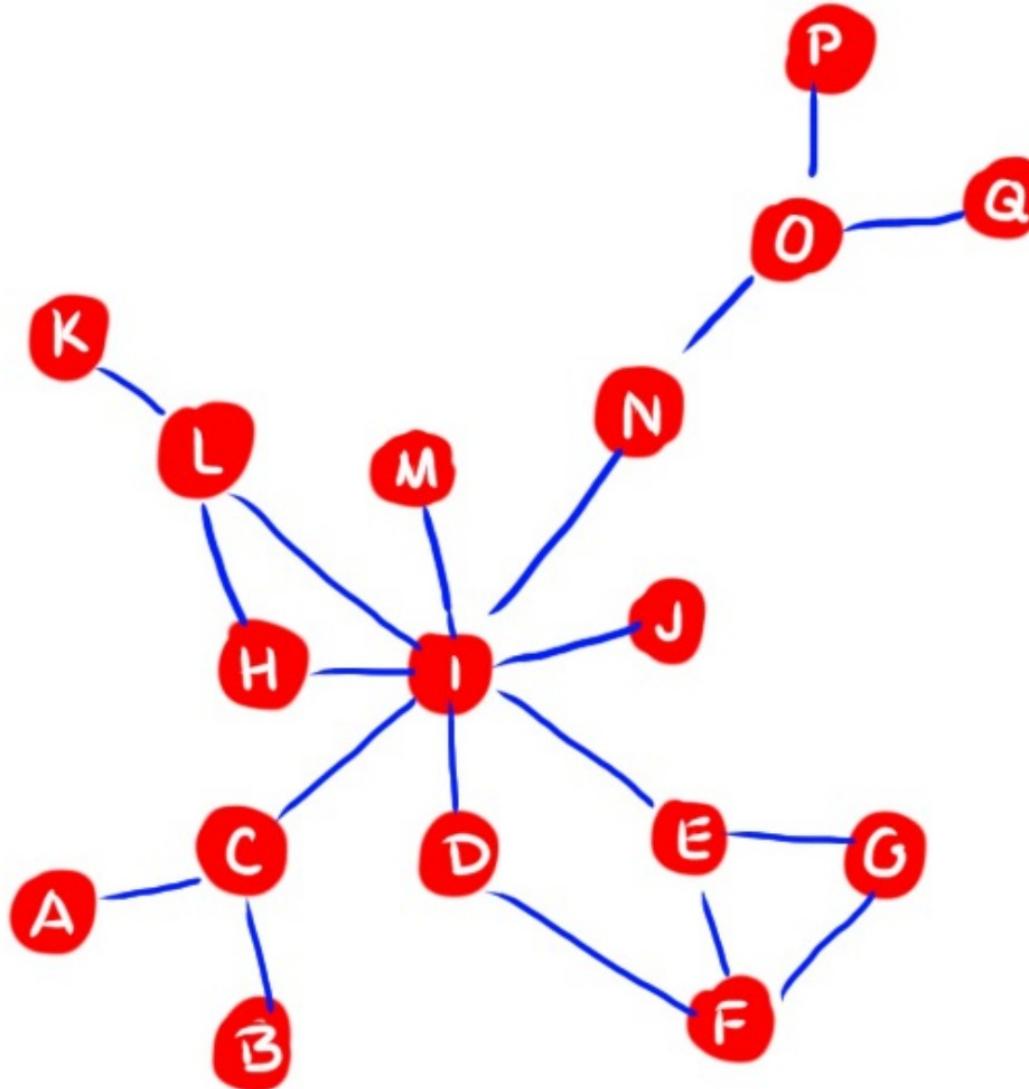
Closeness: Question

- Find a node which has relatively **high degree** but low **closeness**



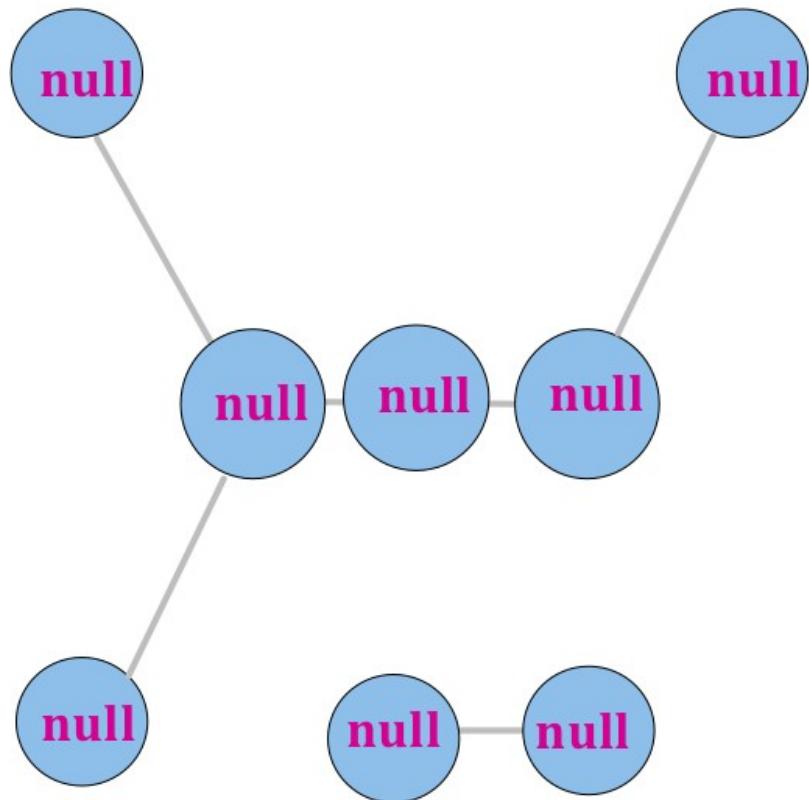
Closeness: Question

- Find a node which has **low degree** but **high closeness**



Closeness: unconnected graph

- What if the graph is **not connected?**



We get null score for all nodes,
if the graph is not connected!

$$C_C(i) = \frac{1}{\sum_{j=1}^N d(i, j)}$$

instead of *null*, we could also interpret it as 0 if *infinity* is the distance between unconnected nodes

Harmonic: Definition

- Replace the average distance with the **harmonic mean** of all distances

Harmonic Centrality:

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i, j)} = \sum_{d(i, j) < \infty, j \neq i} \frac{1}{d(i, j)}$$

- Strongly correlated to closeness centrality
- Naturally also accounts for nodes j that cannot reach i
- Can be applied to graphs that are not connected

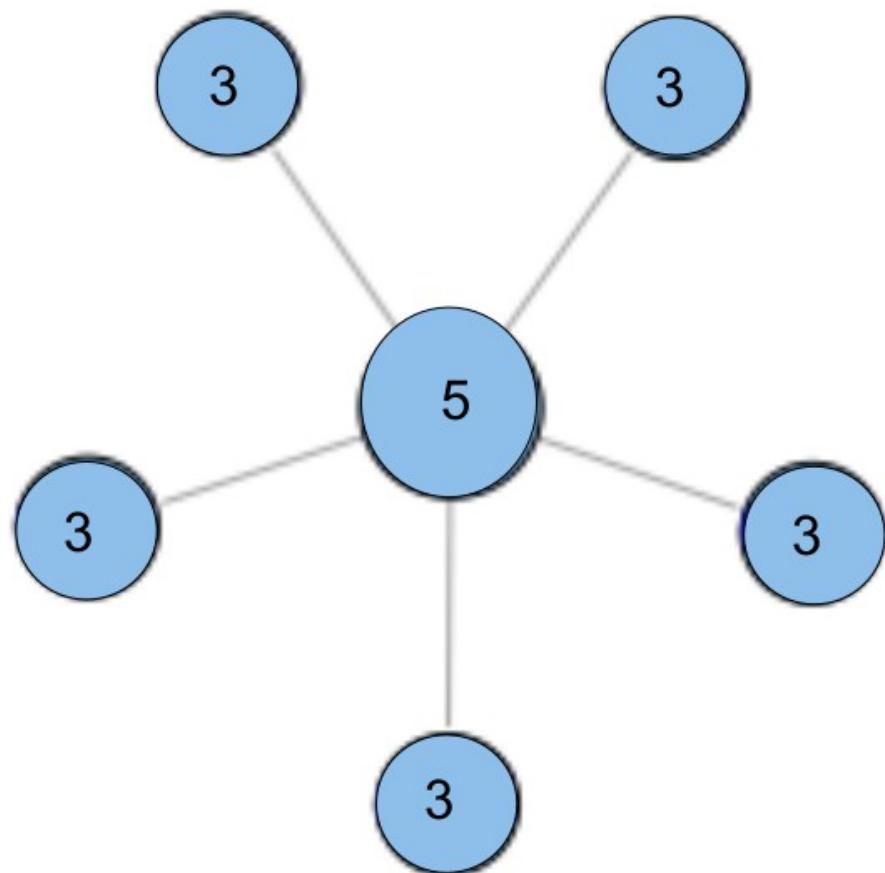
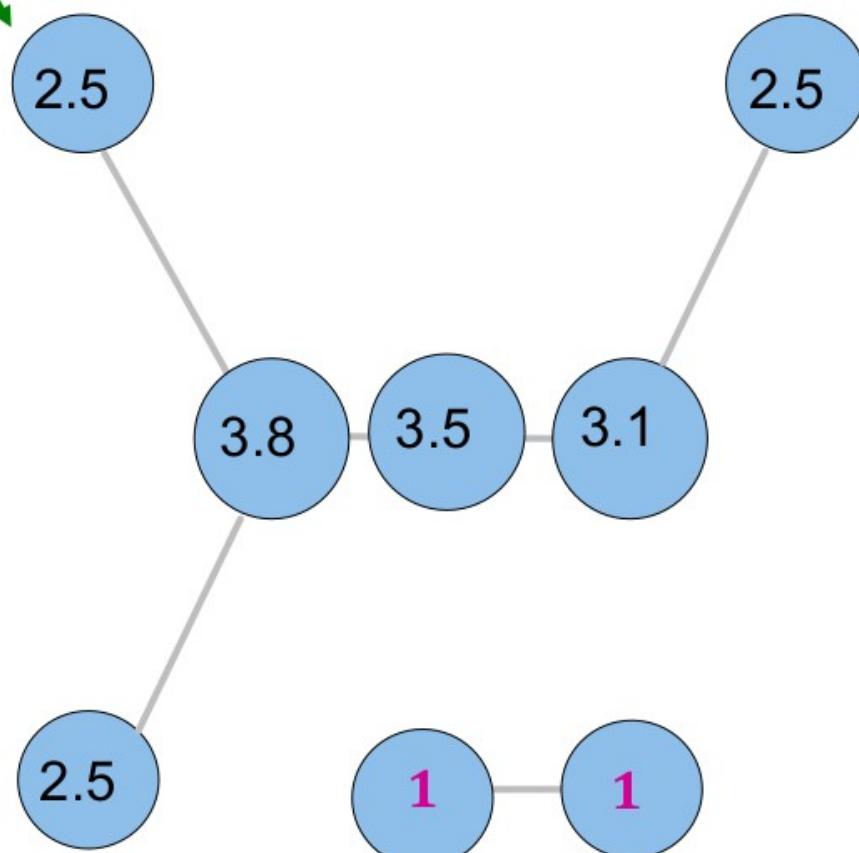
Normalized Harmonic Centrality:

$$C'_H(i) = C_H(i)/(n - 1)$$

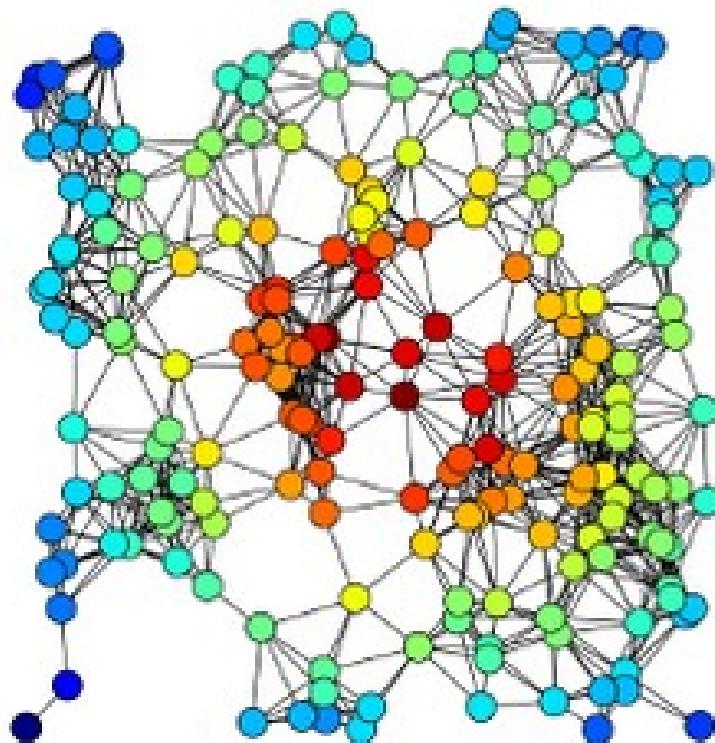
Harmonic: Toy Networks

- Non-normalized version:

$$c_{harm} = \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.5$$

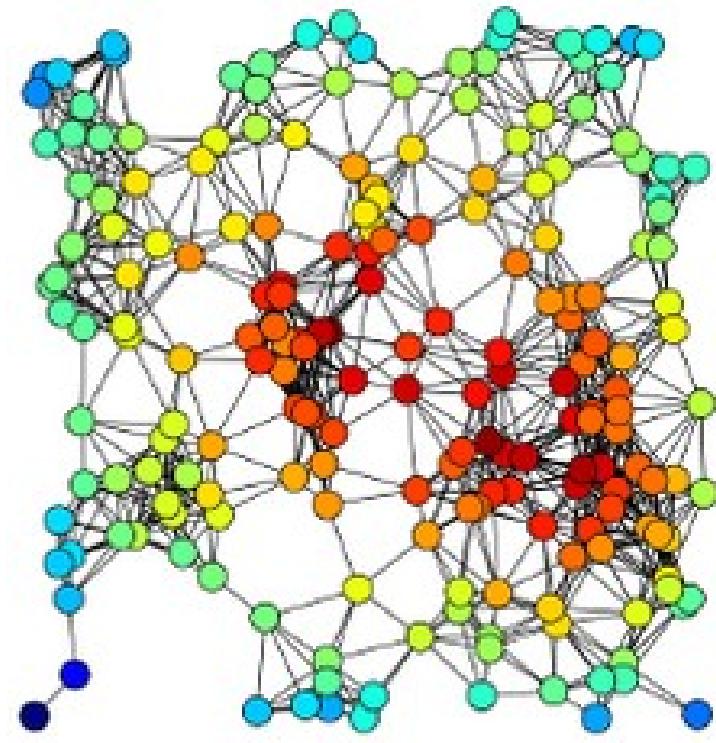


Closeness vs Harmonic



Closeness Centrality

$$C_C(i) = \frac{1}{\sum_{j=1}^N d(i, j)}$$

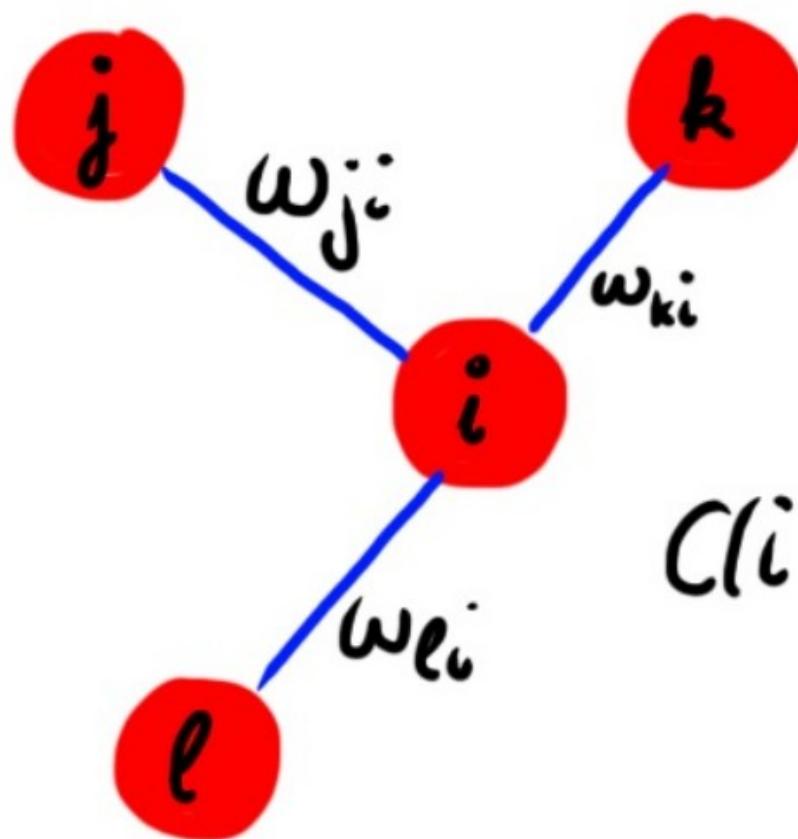


Harmonic Centrality

$$C_H(i) = \sum_{j \neq i} \frac{1}{d(i, j)}$$

Eigenvector Centrality

- How “central” you are depends on how “central” your neighbors are



$$C(i) = w_{ji} \cdot C(j) + w_{ki} \cdot C(k) + w_{li} \cdot C(l)$$

Eigenvector Centrality

Eigenvector Centrality:

$$C_E(i) = \frac{1}{\lambda} \sum_{j=1}^n A_{ji} \times C_E(j)$$

where λ is a constant and

A_{ij} the adjacency matrix (1 if (i,j) are connected, 0 otherwise)

(with a small rearrangement) this can we rewritten in vector notation as in the eigenvector equation

$$Ax = \lambda x$$

where x is the eigenvector, and its i -th component is the centrality of node i

In general, there will be many different eigenvalues λ for which a non-zero eigenvector solution exists. However, the additional requirement that all the entries in the eigenvector be non-negative implies (by the Perron–Frobenius theorem) that only the greatest eigenvalue results in the desired centrality measure

Bonacich eigenvector centrality

also known as Bonacich Power Centrality

$$c_i(\beta) = \sum (\alpha + \beta c_j) A_{ji}$$

- α is a normalization constant
- β determines how important the centrality of your neighbors is
- A is the adjacency matrix (can be weighted)

Bonacich eigenvector centrality

also known as Bonacich Power Centrality

small $\beta \rightarrow$ high attenuation

only your immediate friends matter, and their importance is factored in only a bit

high $\beta \rightarrow$ low attenuation

global network structure matters (your friends, your friends' of friends etc.)

$\beta = 0$ yields simple degree centrality

$$c_i(\beta) = \sum_j (\alpha \boxed{}) A_{ji}$$

Eigenvector Variants

- There are other **variants** of eigenvector centrality, such as:
 - **PageRank**
 - (normalized eigen vector + random jumps)
[we will talk in detail about that later]
 - **Katz Centrality**
 - (connections with distant neighbors are penalized)

$$C_{\text{Katz}}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^n \alpha^k (A^k)_{ji}$$

Centrality in Directed Networks

- **Degree:**
 - in and out centrality

- **Betweenness:**

- Consider only directed paths: $C_B(i) = \sum_{j \neq k} \frac{g_{jk}(i)}{g_{jk}}$
- When normalizing take care of ordered pairs

$$C'_B(i) = \frac{C_B(i)}{(n-1)(n-2)}$$

number of ordered pairs is
2x the number of unordered

- **Closeness**

- Consider only directed paths

- **Eigenvector** (already prepared)

Centrality in Weighted Networks

- **Degree:**
 - Sum weights (*non-weighted equals weight=1 for all edges*)
- **Betweenness and Closeness:**
 - Consider weighted distance
- **Eigenvector**
 - Consider weighted adjacency matrix

Node Centralities: Conclusion

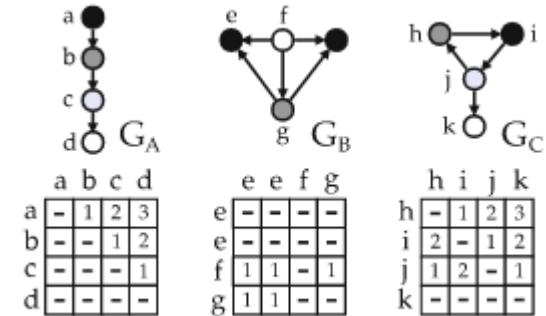
- There are other node centrality metrics, but these are the “**quintessential**”

Finding Dominant Nodes Using Graphlets

David Aparício^(✉), Pedro Ribeiro, Fernando Silva, and Jorge Silva

CRACS & INESC-TEC and the Department of Computer Science,
Faculty of Sciences, University of Porto, 4169-007 Porto, Portugal
`{daparicio,pribeiro,fds}@dcc.fc.up.pt, jorge.m.silva@inesctec.pt`

$$D(o) = \left(\lambda \times \sum_{o_i \in \mathcal{I}(o)} \beta^{k-d(o,o_i)} \right) - \left((1-\lambda) \times \sum_{o_j \in \mathcal{S}(o)} \beta^{k-d(o_j,o)} \right)$$

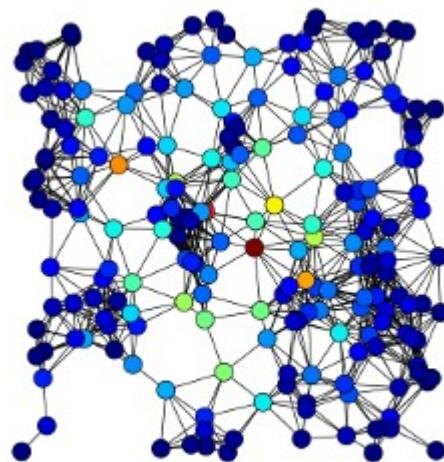


A subgraph-based ranking system for professional tennis players

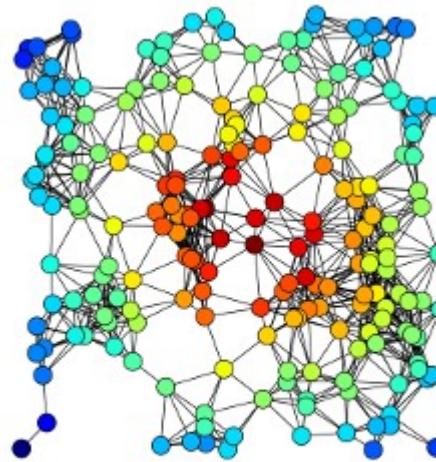
David Aparício, Pedro Ribeiro and Fernando Silva

- Which one to use depends on **what you want to achieve or measure**
 - Worry about understanding the concepts
 - They enlarge your graph vocabulary

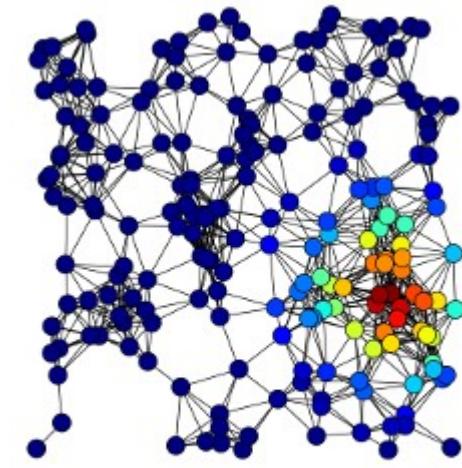
Node Centralities: Conclusion



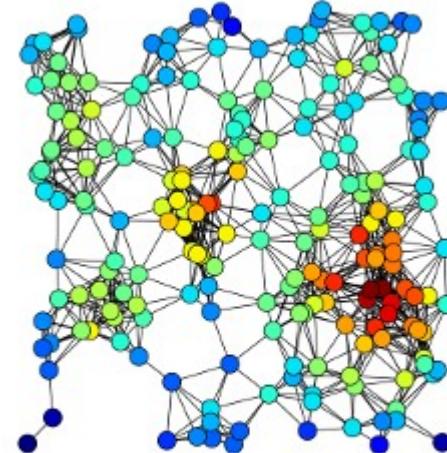
Betweenness



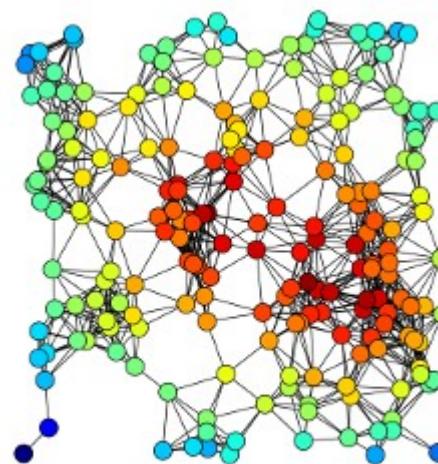
Closeness



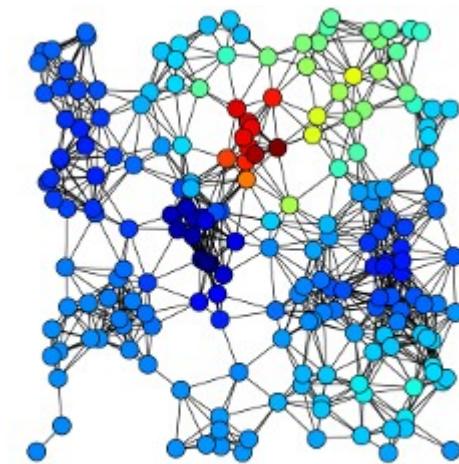
Eigenvector



Degree



Harmonic



Katz

Node Centralities: Conclusion

- All (major) network analysis packages provide them:



The #1 Database for Connected Data

Centrality algorithms are used to determine includes the following centrality algorithms

- Production-quality
 - Page Rank
 - Betweenness Centrality
- Alpha
 - ArticleRank
 - Closeness Centrality
 - Harmonic Centrality
 - Degree Centrality
 - Eigenvector Centrality
 - HITS



Centrality

Degree

<code>degree_centrality (G)</code>	Compute the degree centrality for nodes.
<code>in_degree_centrality (G)</code>	Compute the in-degree centrality for nodes.
<code>out_degree_centrality (G)</code>	Compute the out-degree centrality for nodes.

Eigenvector

<code>eigenvector_centrality (G[, max_iter, tol, ...])</code>	Compute the eigenvector centrality for the graph G.
<code>eigenvector_centrality_numpy (G[, weight, ...])</code>	Compute the eigenvector centrality for the graph G.
<code>katz_centrality (G[, alpha, beta, max_iter, ...])</code>	Compute the Katz centrality for the nodes of the graph G.
<code>katz_centrality_numpy (G[, alpha, beta, ...])</code>	Compute the Katz centrality for the graph G.

Closeness

<code>closeness_centrality (G[, u, distance, ...])</code>	Compute closeness centrality for nodes.
<code>incremental_closeness_centrality (G[, edge[, ...]])</code>	Incremental closeness centrality for nodes.

Current Flow Closeness

<code>current_flow_closeness_centrality (G[, ...])</code>	Compute current-flow closeness centrality for nodes.
<code>information_centrality (G[, weight, dtype, ...])</code>	Compute current-flow closeness centrality for nodes.

(Shortest Path) Betweenness

<code>betweenness_centrality (G[, k, normalized, ...])</code>	Compute the shortest-path betweenness centrality for r
---	--



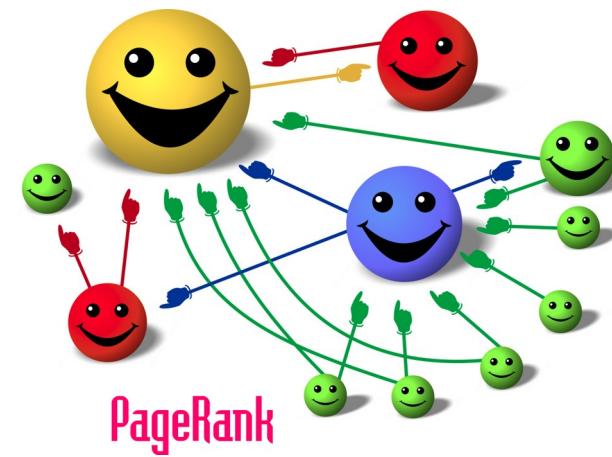
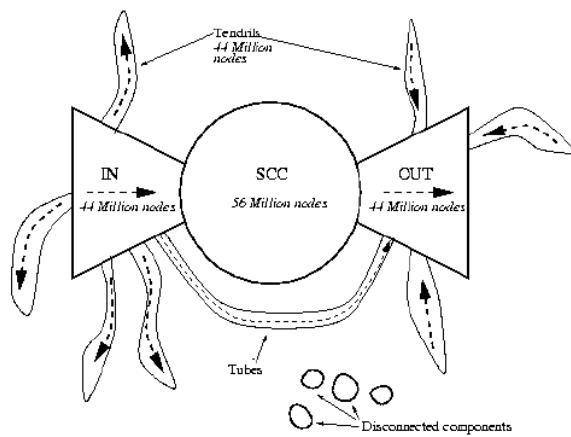
8. Centrality Measures

- 8.1. `igraph_closeness` — Closeness centrality calculations for some vertices.
- 8.2. `igraph_harmonic_centrality` — Harmonic centrality for some vertices.
- 8.3. `igraph_betweenness` — Betweenness centrality of some vertices.
- 8.4. `igraph_edge_betweenness` — Betweenness centrality of the edges.
- 8.5. `igraph_pagerank_algo_t` — PageRank algorithm implementation
- 8.6. `igraph_pagerank` — Calculates the Google PageRank for the specified vertices.
- 8.7. `igraph_personalized_pagerank` — Calculates the personalized Google PageRank for the specified vertices.
- 8.8. `igraph_personalized_pagerank_vs` — Calculates the personalized Google PageRank for the specified vertices.
- 8.9. `igraph_constraint` — Burt's constraint scores.
- 8.10. `igraph_maxdegree` — The maximum degree in a graph (or set of vertices).
- 8.11. `igraph_strength` — Strength of the vertices, weighted vertex degree in other words.
- 8.12. `igraph_eigenvector_centrality` — Eigenvector centrality of the vertices

- Also all (major) network analysis and visualization platforms:



Link Analysis: PageRank



(Heavily based on slides from Jure Leskovec and Lada Adamic @ Stanford University)

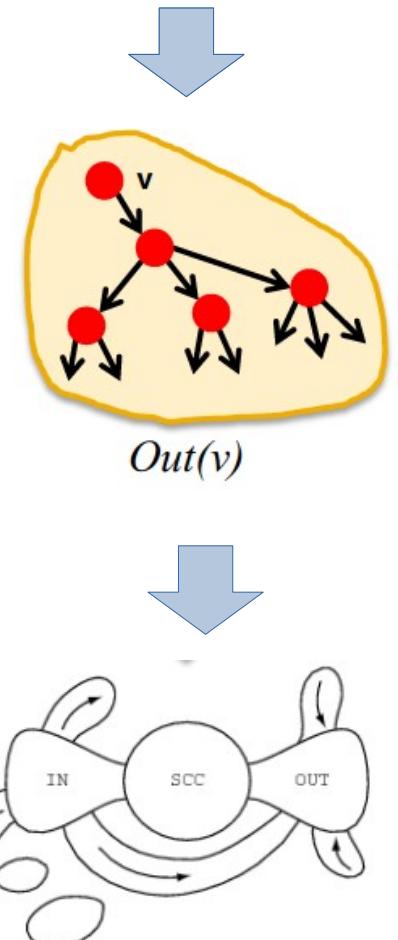
Web as a Graph

Structure of the Web

- On this lecture we will talk about how does the **Web graph** look like:



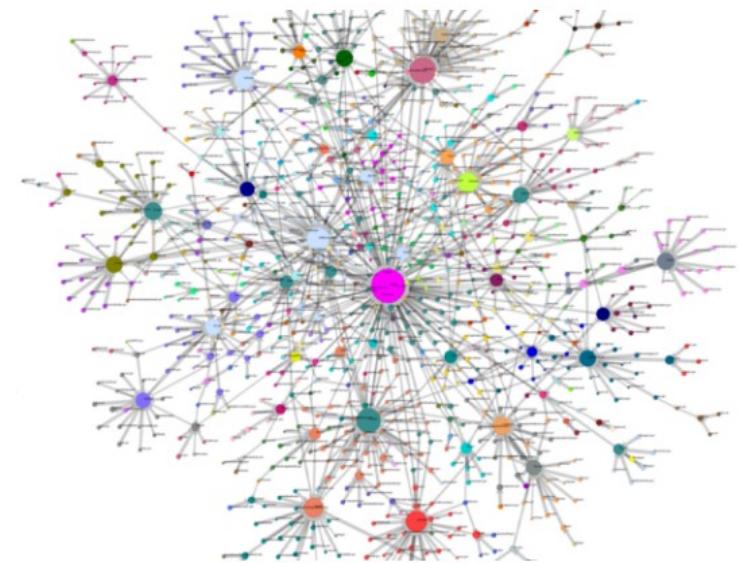
- 1) We will take a real system: **the Web**
- 2) We will represent it as a **directed graph**
- 3) We will use the language of **graph theory**
 - Strongly Connected Components
- 4) We will design a **computational experiment**:
 - Find In-and Out-components of a given node v
- 5) We will learn something about the **structure of the Web: BOWTIE!**



The Web as a Graph

Q: what does the Web “look like” at a global level?

- **Web as a graph:**
 - Nodes = web pages
 - Edges = hyperlinks
- Side issue: **what is a node?**
 - Dynamic pages created on the fly
 - “dark matter” – inaccessible database generated pages



The Web as a Graph: Example

I'm giving
a class on
Network
Science

Classes are
on FC6
building

Computer
Science
Department
at FCUP

University
of
Porto

The Web as a Graph: Example

I'm giving
a class on
Network
Science

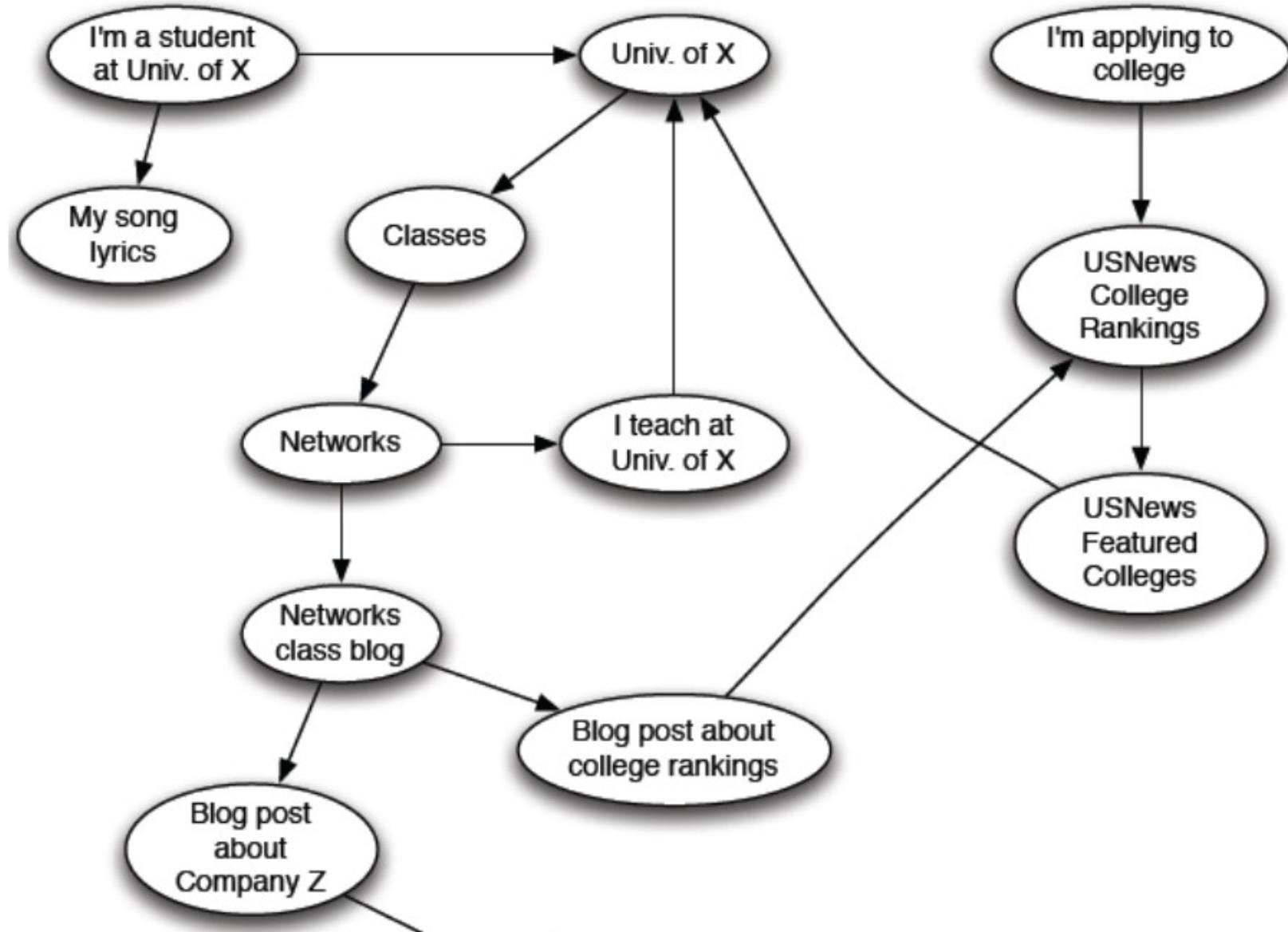
Classes are
on FC6
building

Computer
Science
Department
at FCUP

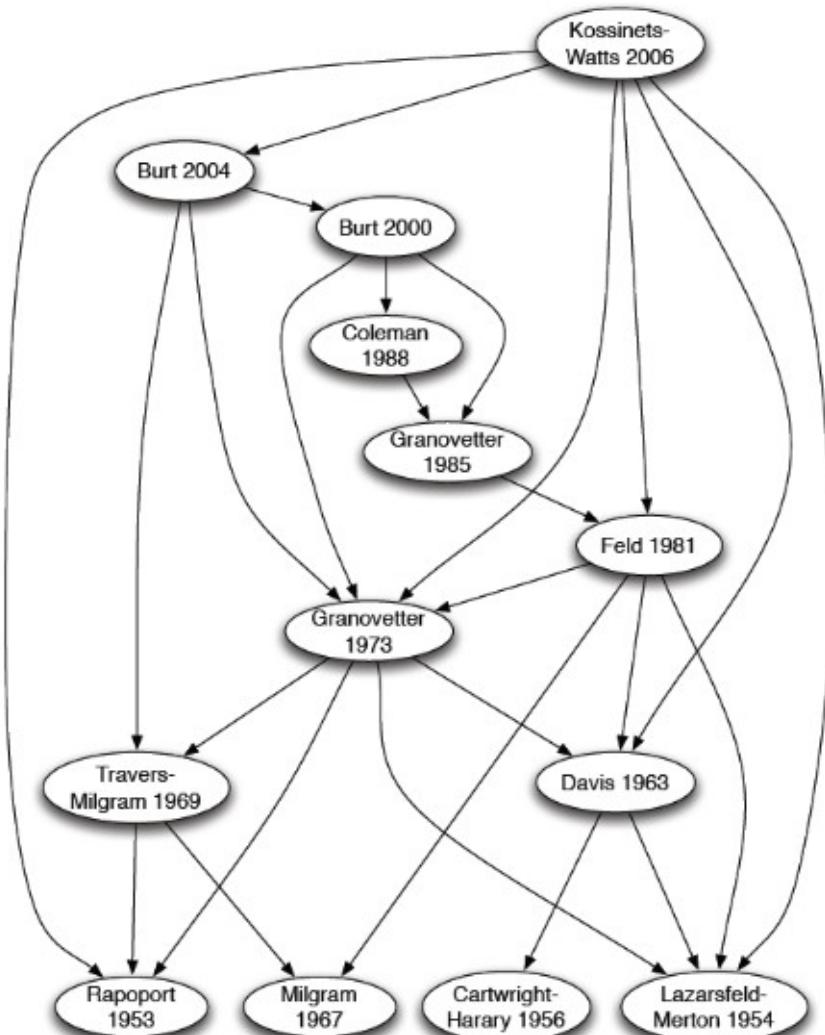
University
of
Porto

- In early days of the Web links were **navigational**
- Today many links are **transactional** (used not to navigate from page to page, but to post, comment, like, buy, ...)

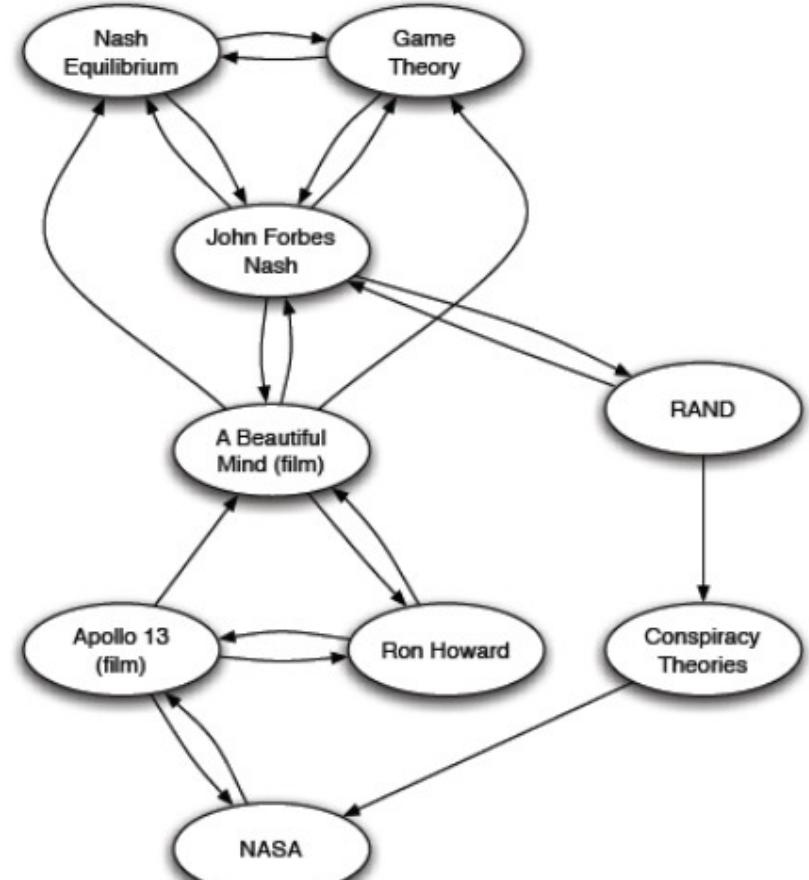
The Web as a Directed Graph



Other Information Networks



Citations



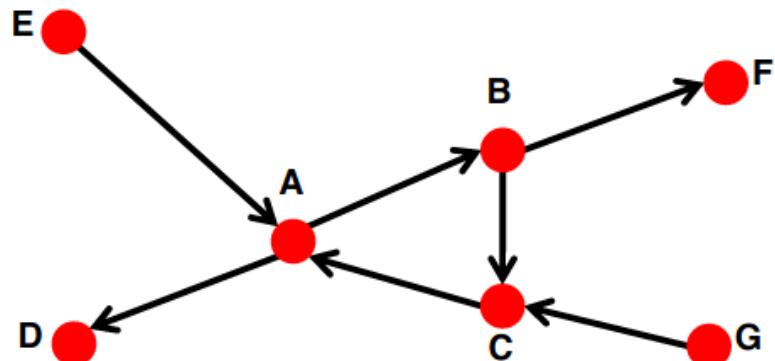
Wikipedia

What does the Web look like?

- How is the Web linked?
- What is the “map” of the Web?

Web as a **directed graph** [Broder et al. 2000]:

- Given node v , what nodes can v reach?
- What other nodes can reach v ?



$$In(v) = \{w \mid w \text{ can reach } v\}$$

$$Out(v) = \{w \mid v \text{ can reach } w\}$$

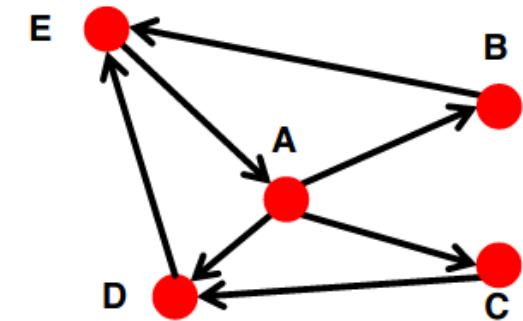
For example:
 $In(A) = \{A,B,C,E,G\}$
 $Out(A)=\{A,B,C,D,F\}$

Reasoning About Directed Graphs

- Two types of directed graphs:

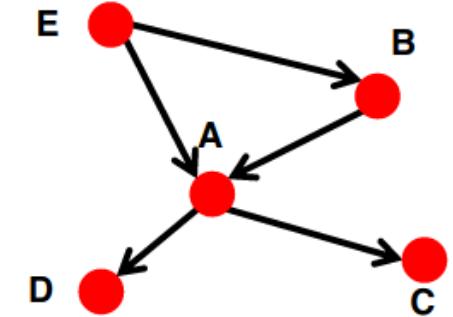
- **Strongly connected:**

- Any node can reach any node via a directed path
 $In(A)=Out(A)=\{A,B,C,D,E\}$



- **Directed Acyclic Graph (DAG):**

- Has no cycles: if u can reach v , then v cannot reach u

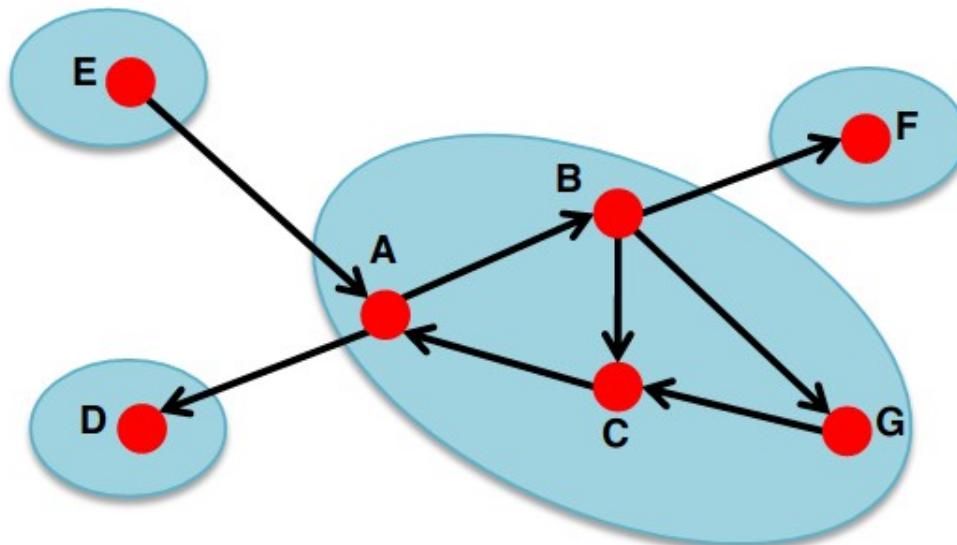


- **Any directed graph (the Web) can be expressed in terms of these two types!**

- Is the Web a big strongly connected graph or a DAG?

Strongly Connected Component

- A **Strongly Connected Component** (SCC) is a set of nodes S so that:
 - Every pair of nodes in S can reach each other
 - There is no larger set containing S with this property



Strongly connected components of the graph:
 $\{A,B,C,G\}$, $\{D\}$, $\{E\}$, $\{F\}$

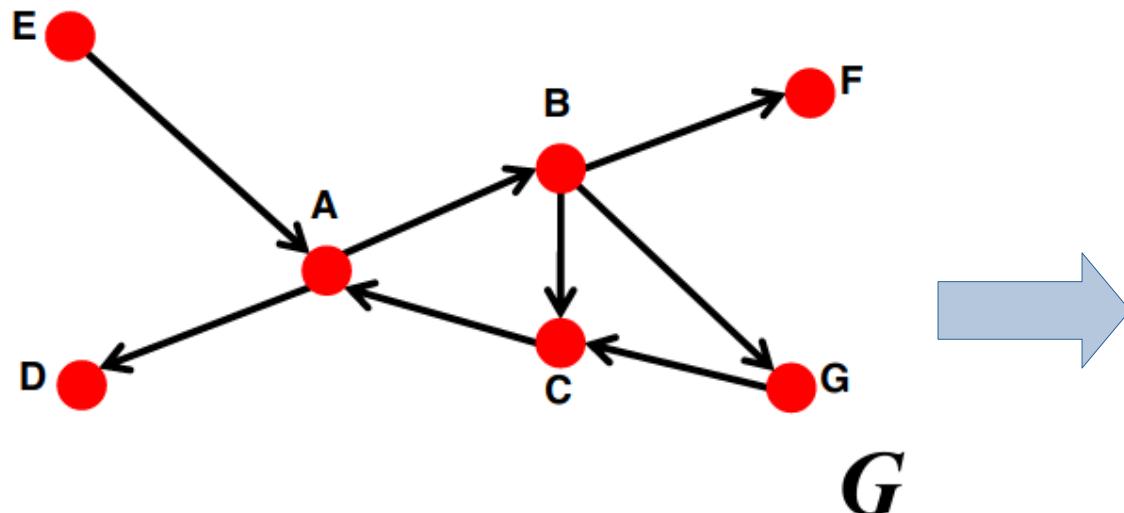
Strongly Connected Component

- Fact: Every directed graph is a DAG on its SCCs

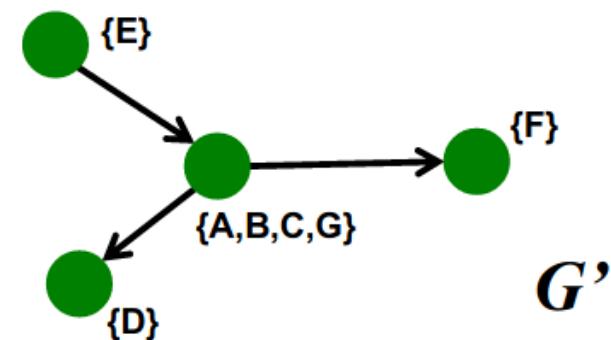
1)SCCs partition the nodes of **G**

- That is, each node is in exactly one SCC

2)If we build a graph **G'** whose nodes are SCCs, and with an edge between nodes of **G'** if there is an edge between corresponding SCCs in **G**, then **G'** is a DAG



(1) Strongly connected components of graph **G**: $\{A, B, C, G\}$, $\{D\}$, $\{E\}$, $\{F\}$
(2) **G'** is a DAG:



Structure of the Web

- **Broder et al.:** Altavista web crawl (Oct '99)
 - Web crawl is based on a large set of starting points accumulated over time from various sources, including voluntary submissions.
 - 203 million URLs and 1.5 billion links

Goal: Take a large snapshot of the Web and try to understand how its SCCs “fit together” as a DAG

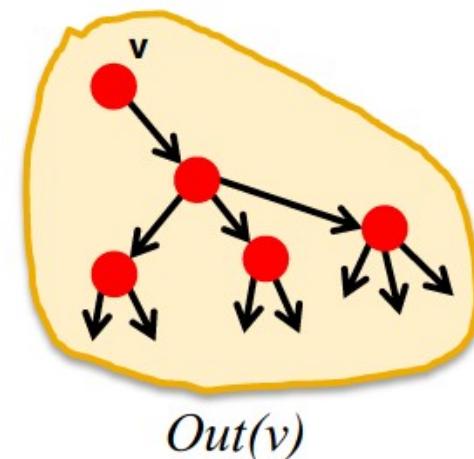


Tomkins,
Broder, and
Kumar

Graph Structure of the Web

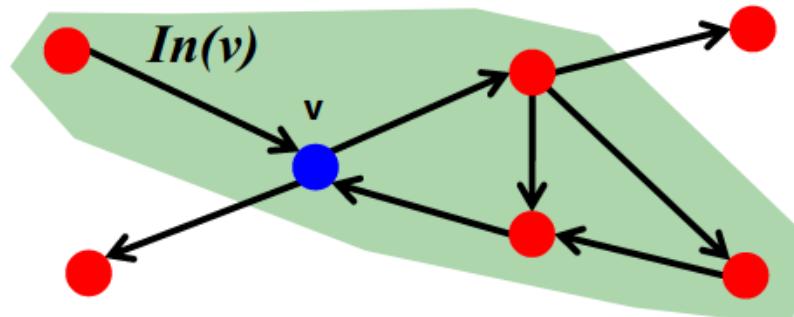
- **Computational issue:**

- Want to find a SCC containing node v ?



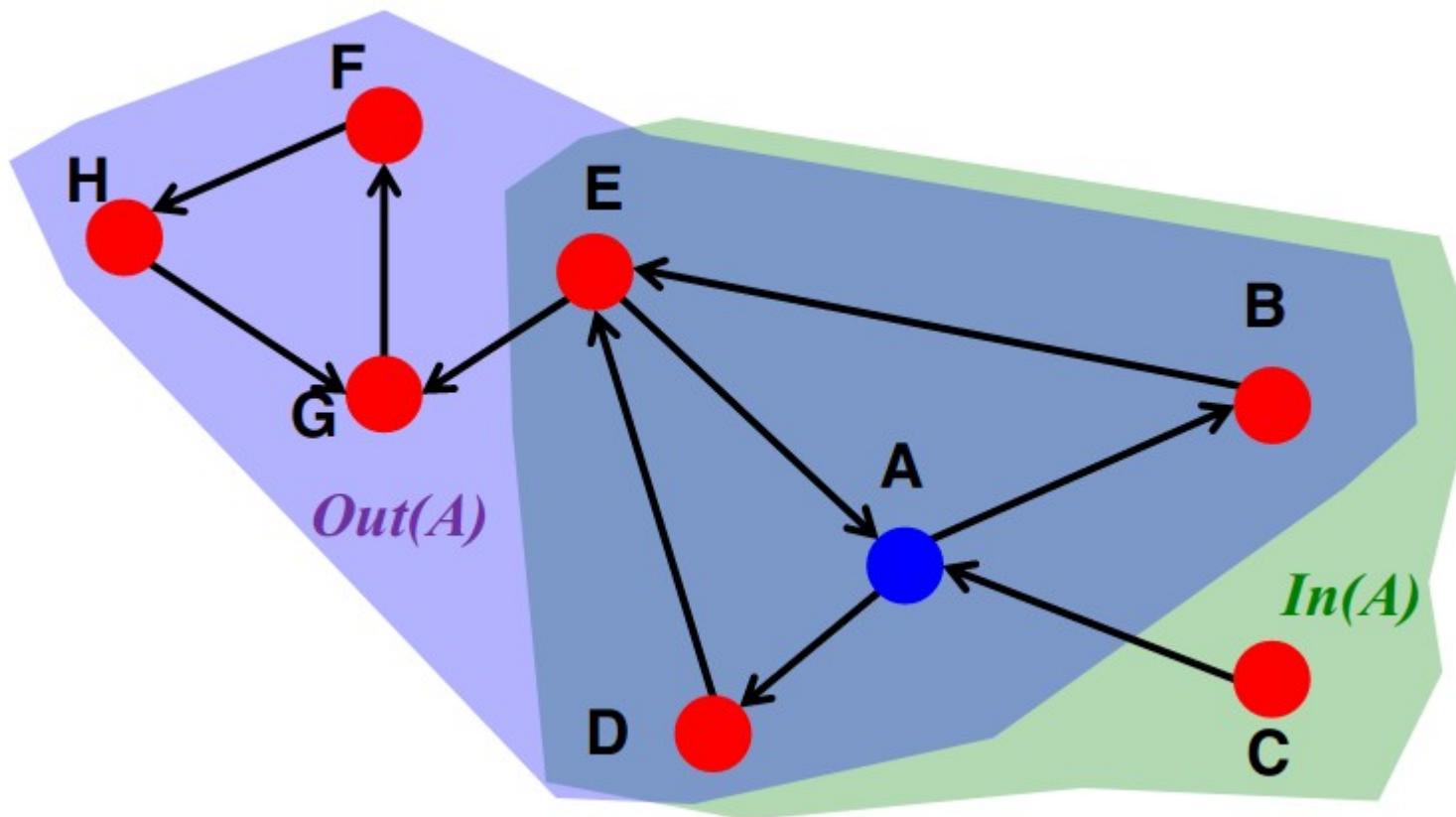
- **Observation:**

- $Out(v)$... nodes that can be reached from v (w/BFS)
 - SCC containing v is:
 $Out(v) \cap In(v) = Out(v, G) \cap Out(v, G')$,
where G' is G with all edge directions flipped



$Out(v) \cap In(v) = SCC$

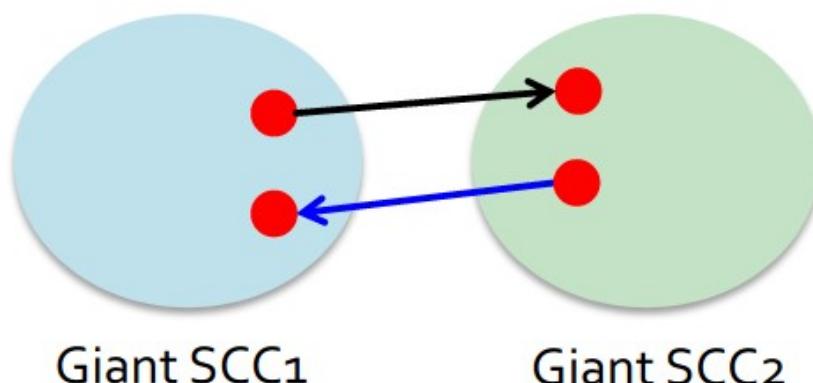
- Example:



- $Out(A) = \{A, B, D, E, F, G, H\}$
- $In(A) = \{A, B, C, D, E\}$
- So, $SCC(A) = Out(A) \cap In(A) = \{A, B, D, E\}$

Graph Structure of the Web

- **There is a single giant SCC**
 - That is, there won't be two SCCs
- **Why only 1 big SCC?** Heuristic argument:
 - Assume two equally big SCCs.
 - It just takes 1 page from one SCC to link to the other SCC.
 - If the two SCCs have millions of pages the likelihood of this not happening is very very small.

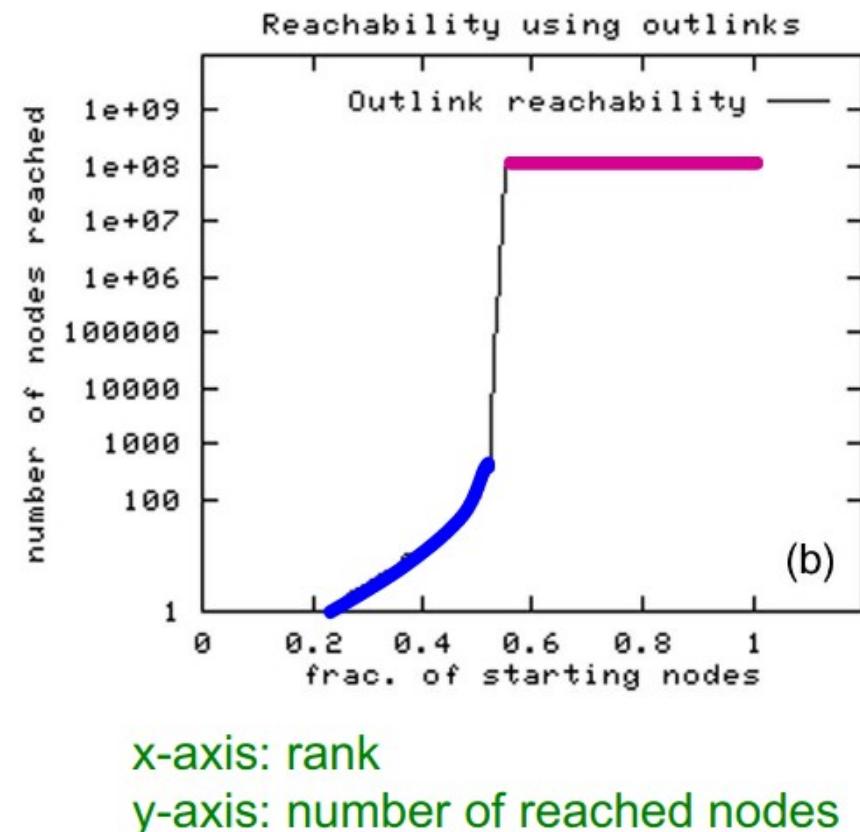


Structure of the Web

- Directed version of the Web graph:
 - Altavista crawl from October 1999
 - 203 million URLs, 1.5 billion links

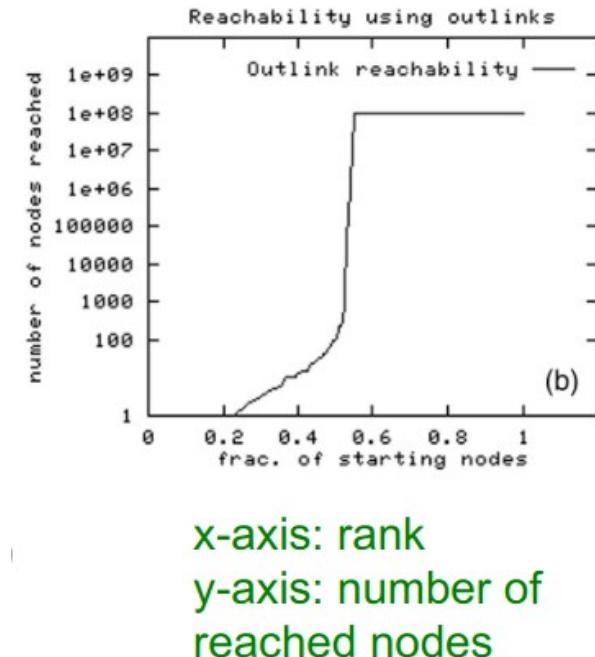
Computation:

- Compute $In(v)$ and $Out(v)$ by starting at random nodes.
- **Observation:** The BFS either visits **many nodes** or **very few**



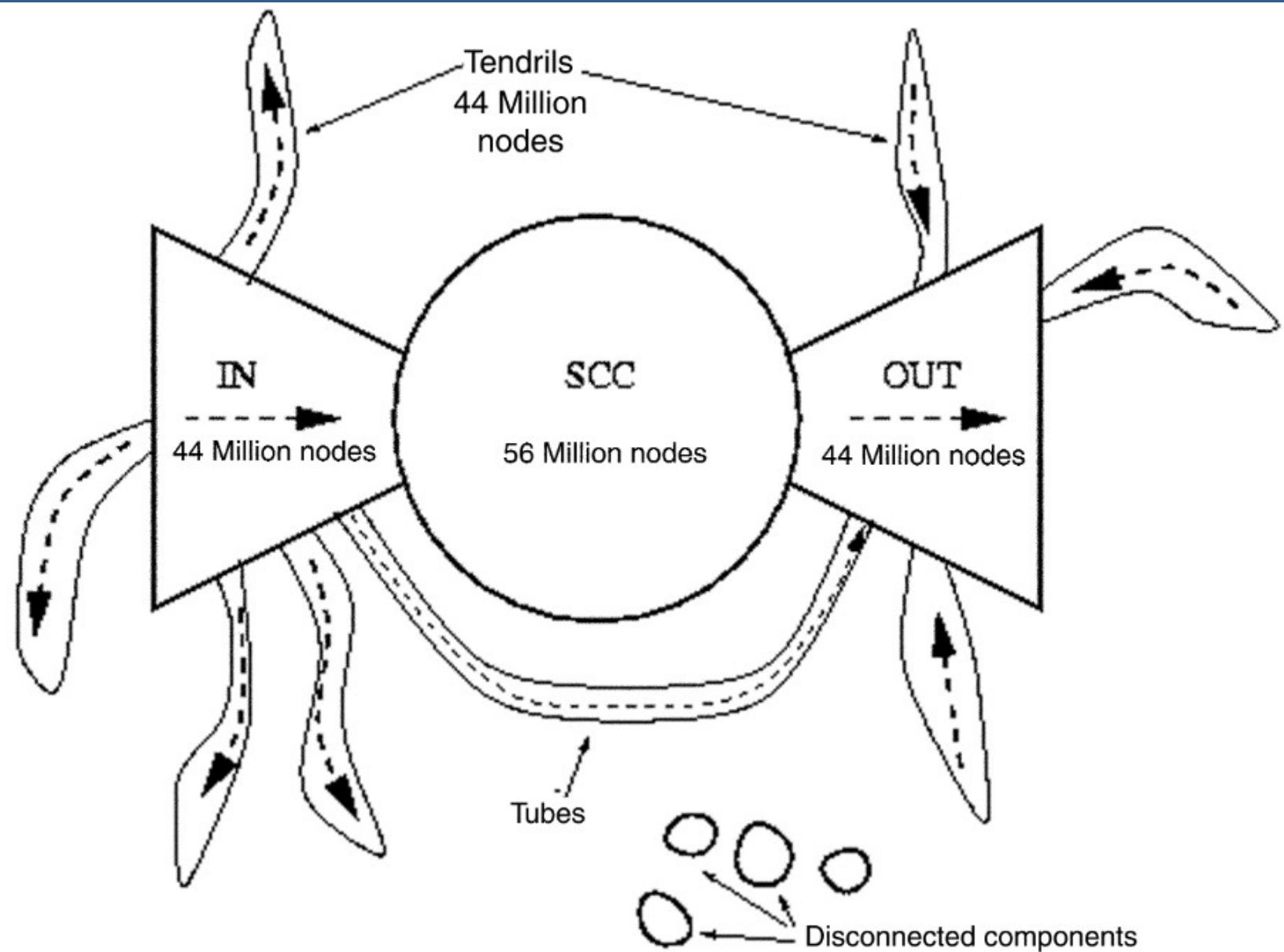
Structure of the Web

- **Result:** Based on IN and OUT of a random node v :
 - $\text{Out}(v) \approx 100$ million (**50%** nodes)
 - $\text{In}(v) \approx 100$ million (**50%** nodes)
 - **Largest SCC:** 56 million (**28%** nodes)



- **What does this tell us about the conceptual picture of the Web Graph?**

Bowtie Structure of the Web



203 million pages, 1.5 billion links [Broder et al. 2000]

How to Organize the Web?

Link Analysis

How to Organize the Web?

- How to organize the Web?

- First try: Human curated

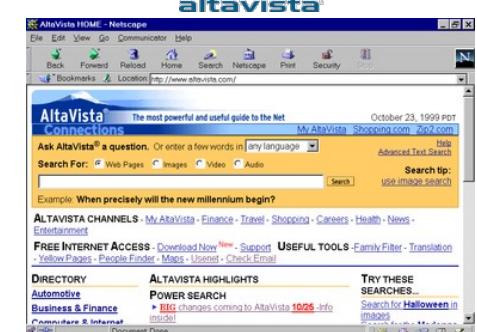
Web directories

- Yahoo, Sapo



- Second try: **Web Search**

- Information Retrieval:** attempts to find relevant docs in a small and trusted set
 - Newspaper articles, Patents, etc.
 - But:** Web is huge, full of untrusted documents, random things, spam, etc.
 - So we need a good way to rank webpages!**



Web Search: Challenges

2 challenges of web search

1) Web contains many sources of information

Who to “trust”?

- **Insight:** Trustworthy pages may point to each other!

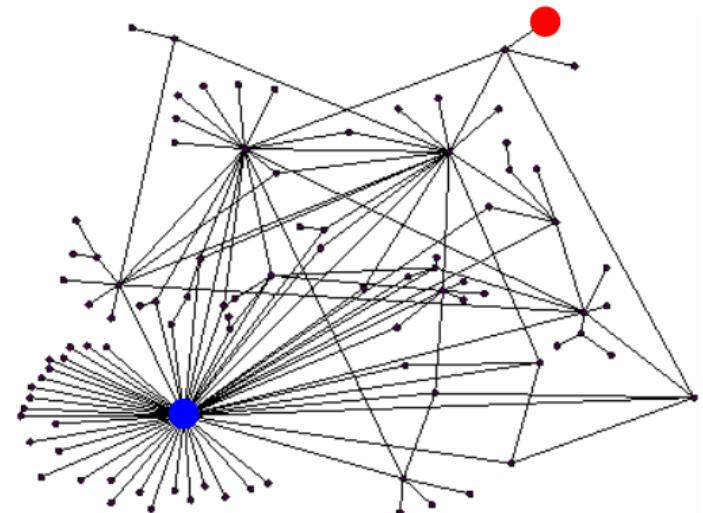
2) What is the “best” answer to query “newspaper”

- No single right answer

- **Insight:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

- Web pages are not equally “important”
 - www.joe-nobody.com vs www.up.pt
- We already know: There is a large diversity in the web graph node connectivity
- **So, let's rank the pages using the web graph link structure!**



Link Analysis Algorithms

- We will cover the following **Link Analysis** approaches to computing the importance of nodes in a graph:
 - Hubs and Authorities (**HITS**)
 - **PageRank**
 - Topic-Specific (**Personalized**) **PageRank**

Sidenote: Various notions of **node centrality**: Node u

- ❑ **Degree centrality** = degree of u
- ❑ **Betweenness centrality** = #shortest paths passing through u
- ❑ **Closeness centrality** = avg. length of shortest paths from u to all other nodes of the network
- ❑ **Eigenvector centrality** = like PageRank

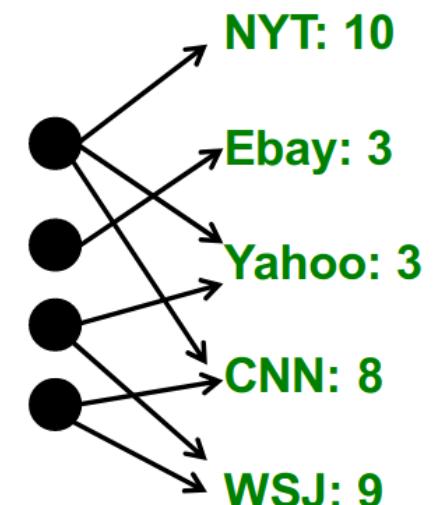
Hubs and Authorities (HITS)

Link Analysis

- Goal(back to the newspaper example):
 - Don't just find newspapers. Find “experts” - pages that link in a coordinated way to good newspapers
- Idea: **Links as votes**
 - Page is more important if it has more links
 - In-coming links? Out-going links?
- Hubs and Authorities

Each page has 2 scores:

 - Quality as an expert (**hub**):
 - Total sum of votes of pages pointed to
 - Quality as an content (**authority**):
 - Total sum of votes of experts
 - **Principle of repeated improvement**



Hubs and Authorities

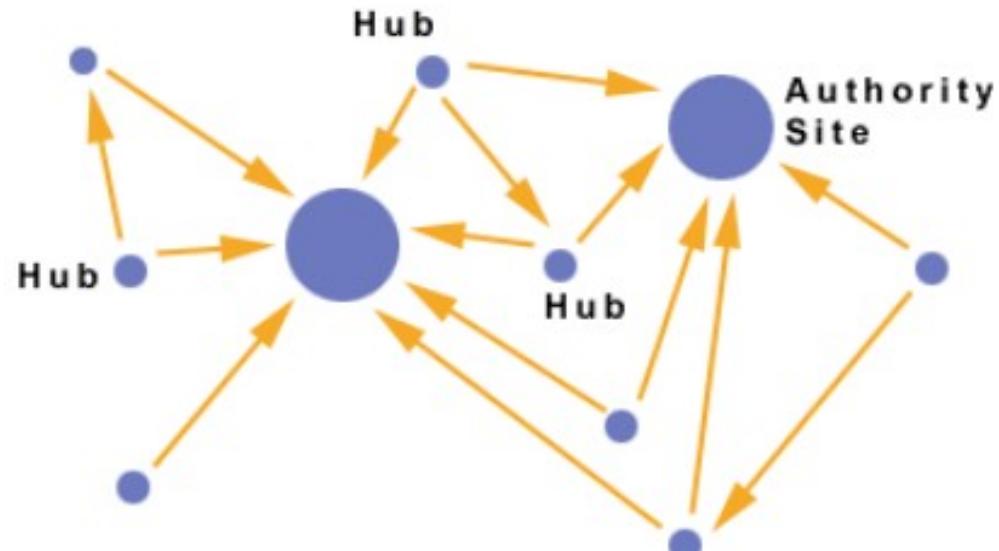
Interesting pages fall into two classes:

1) **Authorities** are pages containing useful information

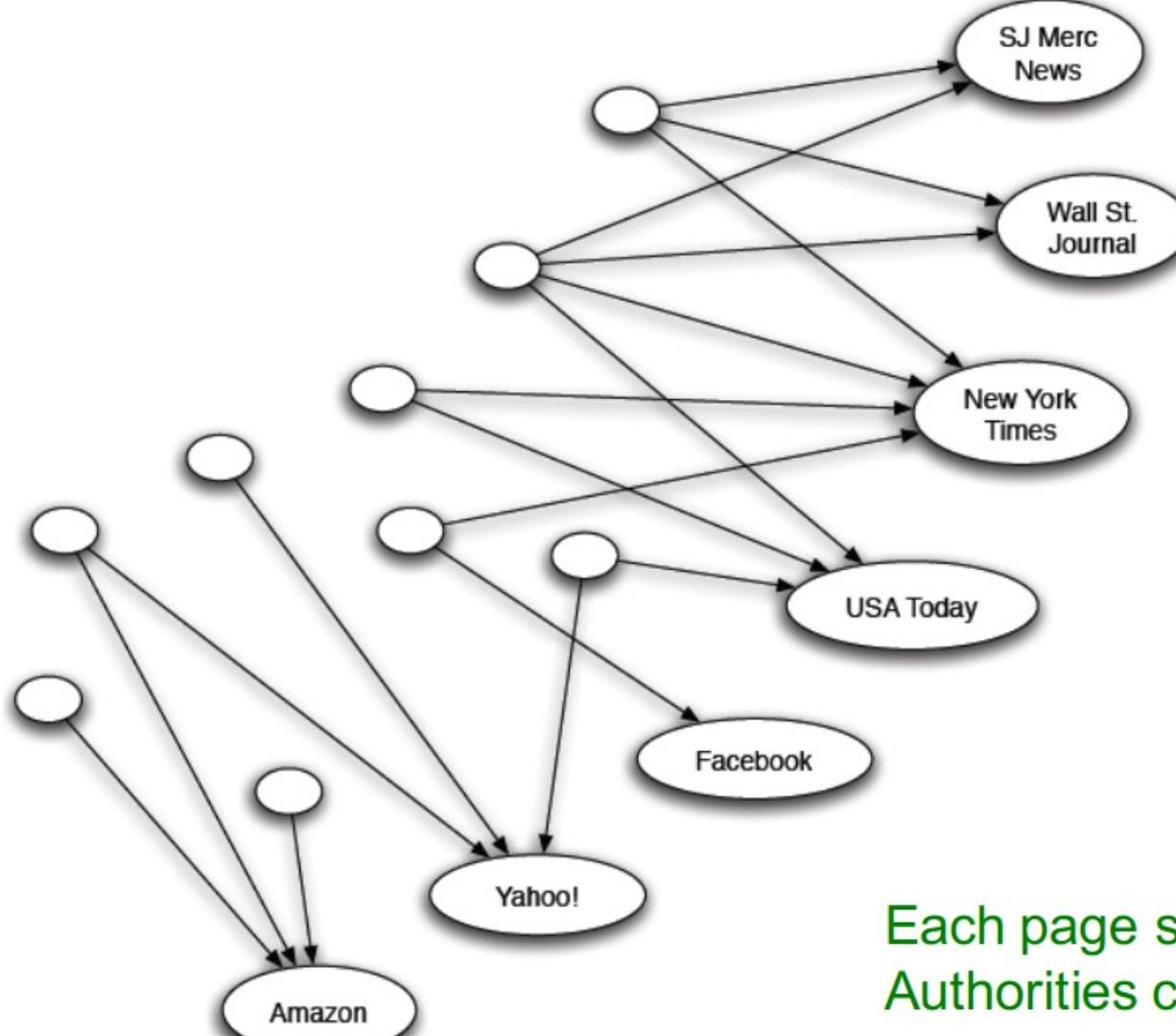
- Newspaper home pages
- Course home pages
- Home pages of auto manufacturers

2) **Hubs** are pages that link to authorities

- List of newspapers
- Course bulletin
- List of auto manufacturers



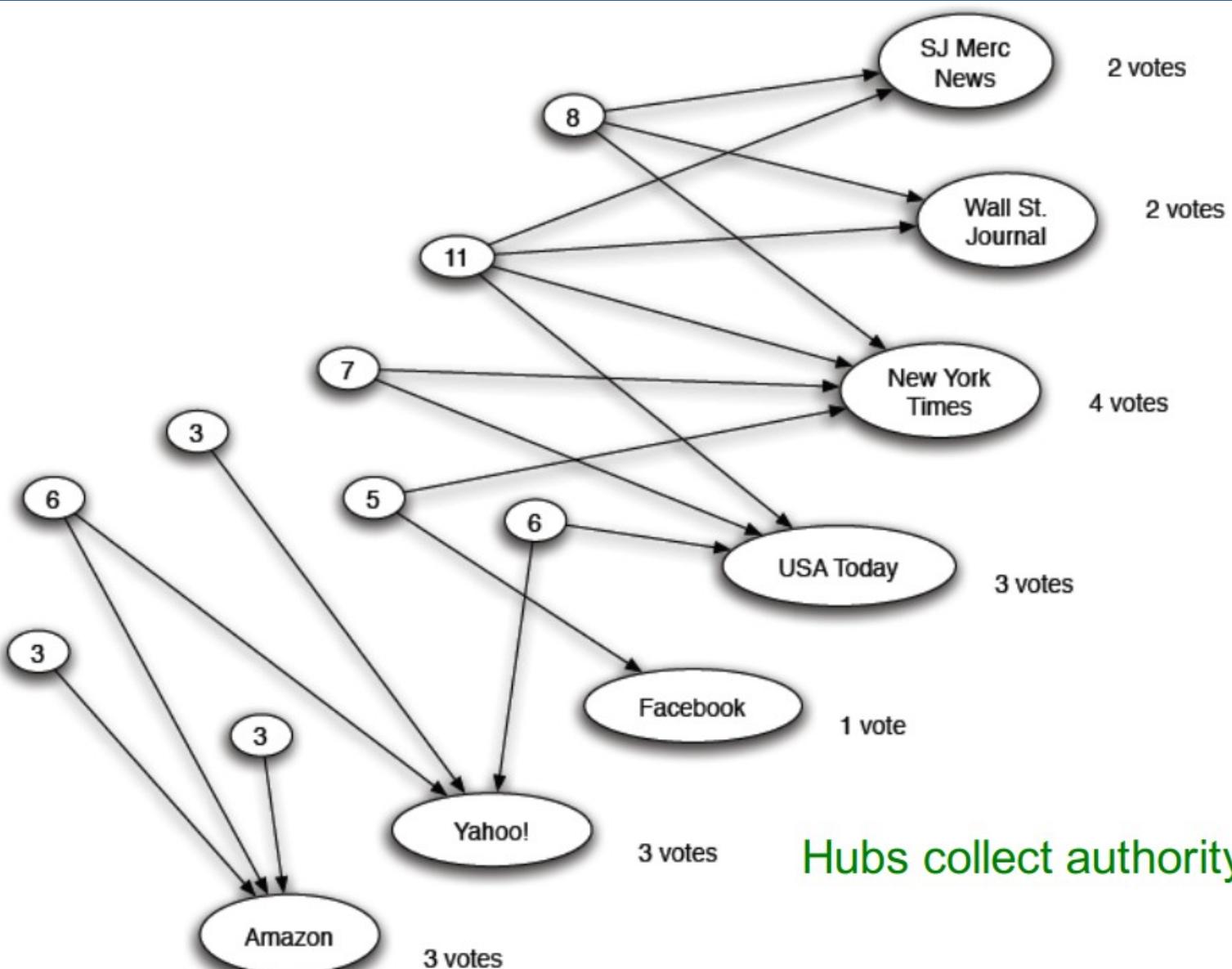
Counting in-links: Authority



Each page starts with **hub score 1**
Authorities collect their votes

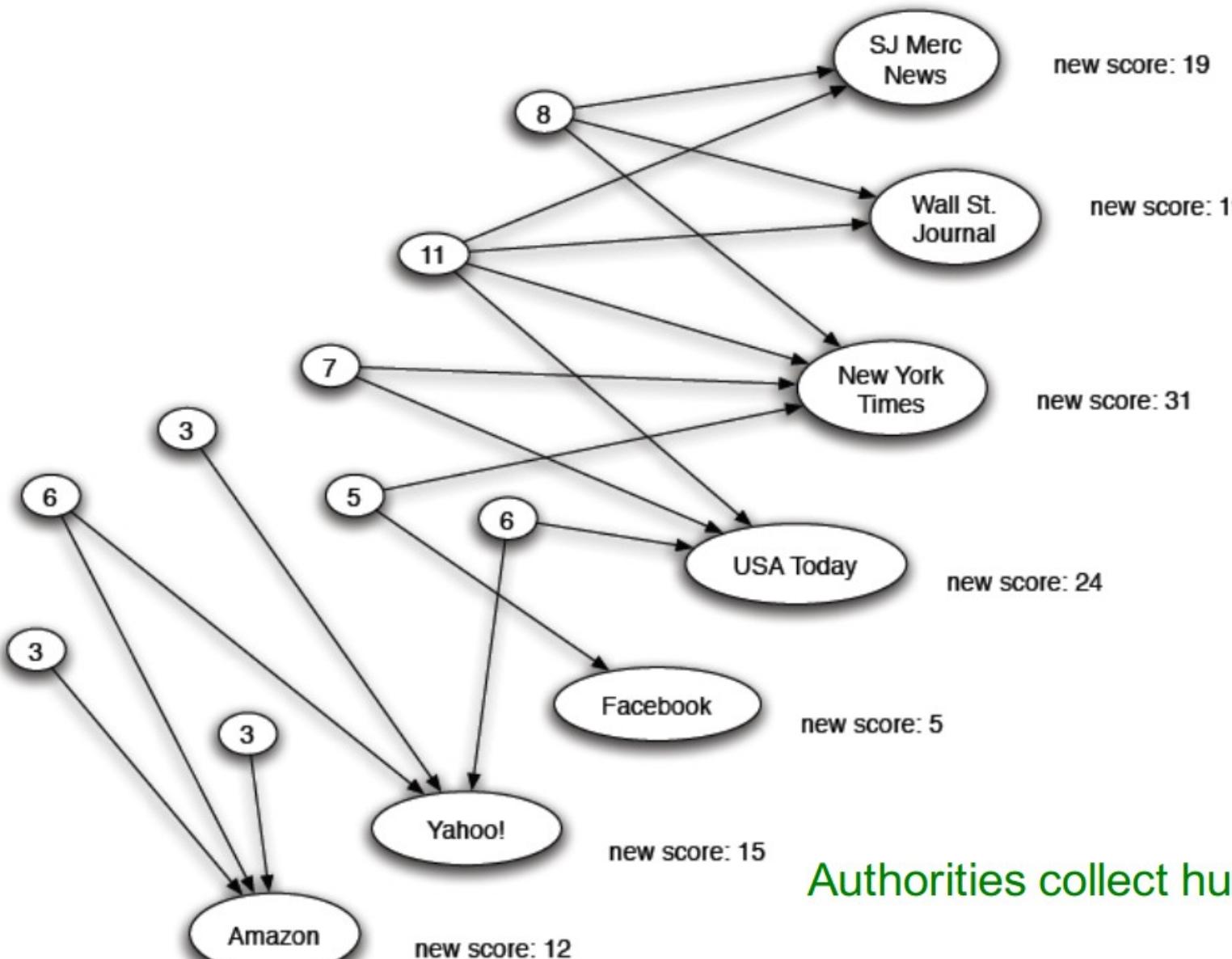
(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and the authority score)

Expert Quality: Hub



(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and authority score)

Reweighting



Authorities collect hub scores

(Note this is idealized example. In reality graph is not bipartite and each page has both a hub and authority score)

Mutually Recursive Definition

- A good **hub** links to many good authorities
- A good **authority** is linked from many good hubs
 - Note a self-reinforcing recursive definition
- Model using two scores for each node:
 - **Hub** score and **Authority** score
 - Represented as vectors h and a , where the i -th element is the hub/authority score of the i -th node

Hubs and Authorities

- **Each page i has 2 scores:**

- Authority score: a_i
 - Hub score: h_i

Convergence criteria:

$$\sum_i \left(h_i^{(t)} - h_i^{(t+1)} \right)^2 < \varepsilon$$

$$\sum_i \left(a_i^{(t)} - a_i^{(t+1)} \right)^2 < \varepsilon$$

HITS algorithm:

- Initialize: $a_j^{(0)} = 1/\sqrt{n}$, $h_j^{(0)} = 1/\sqrt{n}$

- Then keep iterating until **convergence**:

- $\forall i$: Authority: $a_i^{(t+1)} = \sum_{j \rightarrow i} h_j^{(t)}$

- $\forall i$: Hub: $h_i^{(t+1)} = \sum_{i \rightarrow j} a_j^{(t)}$

- $\forall i$: Normalize:

$$\sum_i \left(a_i^{(t+1)} \right)^2 = 1, \sum_j \left(h_j^{(t+1)} \right)^2 = 1$$

■ Hits in the vector notation:

- Vector $a = (a_1 \dots, a_n)$, $h = (h_1 \dots, h_n)$
- Adjacency matrix A ($n \times n$): $A_{ij} = 1$ if $i \rightarrow j$

■ Can rewrite $h_i = \sum_{i \rightarrow j} a_j$ as $h_i = \sum_j A_{ij} \cdot a_j$

■ So: $h = A \cdot a$ And similarly: $a = A^T \cdot h$

■ Repeat until convergence:

- $h^{(t+1)} = A \cdot a^{(t)}$
- $a^{(t+1)} = A^T \cdot h^{(t)}$
- Normalize $a^{(t+1)}$ and $h^{(t+1)}$

Hubs and Authorities

Details

□ What is $a = A^T \cdot h$?

□ Then: $a = A^T \cdot (\underbrace{A \cdot a}_{\text{new } a})$

□ a is updated (in 2 steps):

$$a = A^T (A a) = (A^T A) a$$

□ h is updated (in 2 steps)

$$h = A (A^T h) = (A A^T) h$$

□ Thus, in $2k$ steps:

$$a = (A^T \cdot A)^k \cdot a$$

$$h = (A \cdot A^T)^k \cdot h$$

Repeated matrix powering

- Definition: Eigenvectors & Eigenvalues

- Let $R \cdot x = \lambda \cdot x$
for some scalar λ , vector x , matrix R
 - Then x is an eigenvector, and λ is its eigenvalue

- The steady state (HITS has converged):

- $A^T \cdot A \cdot a = c' \cdot a$
- $A \cdot A^T \cdot h = c'' \cdot h$
- So, **authority** a is eigenvector of $A^T A$
(associated with the largest eigenvalue)
Similarly: **hub** h is eigenvector of AA^T

Note constants c', c''
don't matter as we
normalize them out
every step of HITS

PageRank

(a.k.a., the Google Algorithm)

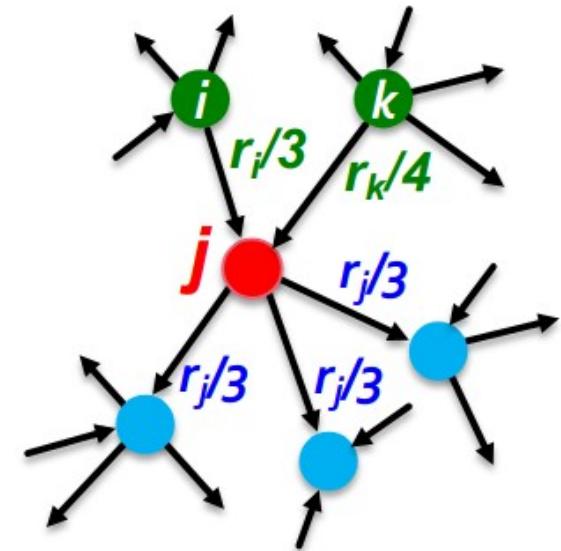
Links as Votes

- **Still the same idea: Links as votes**
 - Page is more important if it has more links
 - In-coming links? Out-going links?
- **Think of in-links as votes:**
 - www.up.pt has 42,000 in-links
 - www.joe-nobody.com has 1 in-link
- **Are all in-links equal?**
 - Links from important pages count more
 - Recursive question!

PageRank: the “Flow” Model

- A “vote” from an important page is worth more:

- Each link’s vote is proportional to the **importance** of its source page
- If page i with importance r_i has d_i out-links, each link gets r_i/d_i votes
- Page j ’s own importance r_j is the sum of the votes on its in-links



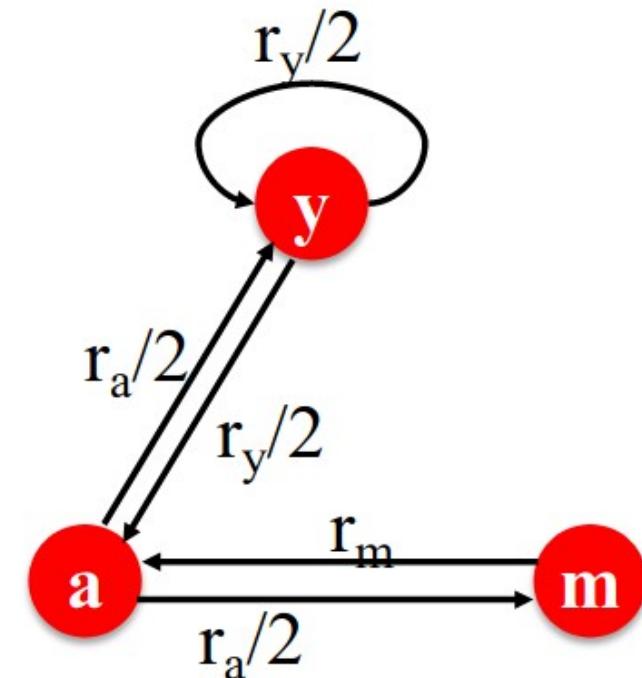
$$r_j = r_i/3 + r_k/4$$

PageRank: the “Flow” Model

- A page is important if it is pointed to by other important pages
- Define a “rank” r_j for node j

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$$

d_i ... out-degree of node i



“Flow” equations:

$$r_y = r_y/2 + r_a/2$$

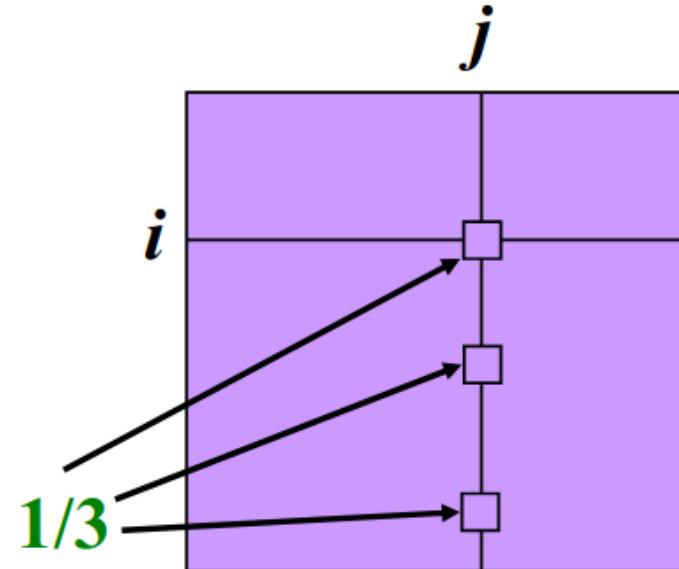
$$r_a = r_y/2 + r_m$$

$$r_m = r_a/2$$

PageRank: Matrix Formulation

■ Stochastic adjacency matrix M

- Let page j have d_j out-links
- If $j \rightarrow i$, then $M_{ij} = \frac{1}{d_j}$
- M is a **column stochastic matrix**
 - Columns sum to 1



■ Rank vector r : An entry per page

- r_i is the importance score of page i
- $\sum_i r_i = 1$

■ The flow equations can be written

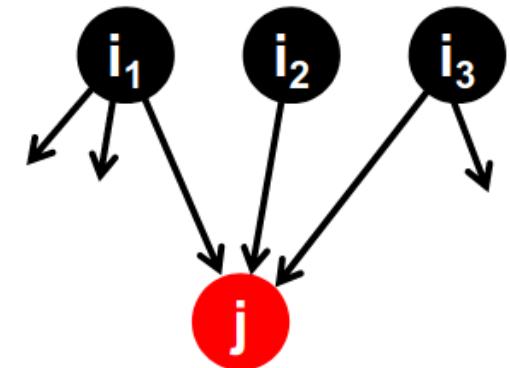
$$r = M \cdot r$$

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$$

Random Walk Interpretation

■ Imagine a random web surfer:

- At any time t , surfer is on some page i
- At time $t + 1$, the surfer follows an out-link from i uniformly at random
- Ends up on some page j linked from i
- Process repeats indefinitely



$$r_j = \sum_{i \rightarrow j} \frac{r_i}{d_{\text{out}}(i)}$$

■ Let:

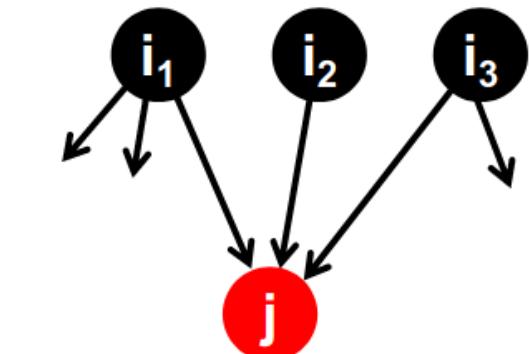
- $p(t)$... vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
- So, $p(t)$ is a probability distribution over pages

The Stationary Distribution

■ Where is the surfer at time $t+1$?

- Follows a link uniformly at random

$$p(t + 1) = M \cdot p(t)$$



$$p(t + 1) = M \cdot p(t)$$

■ Suppose the random walk reaches a state

$$p(t + 1) = M \cdot p(t) = p(t)$$

then $p(t)$ is **stationary distribution** of a random walk

■ Our original rank vector r satisfies $r = M \cdot r$

- So, r is a stationary distribution for the random walk

PageRank

How to Solve?

PageRank: How to Solve?

Given a web graph with n nodes, where the nodes are pages and edges are hyperlinks

- Assign each node an initial page rank
- Repeat until convergence ($\sum_i |r_i^{(t+1)} - r_i^{(t)}| < \epsilon$)
 - Calculate the page rank of each node

$$r_j^{(t+1)} = \sum_{i \rightarrow j} \frac{r_i^{(t)}}{d_i}$$

d_i out-degree of node i

PageRank: How to Solve?

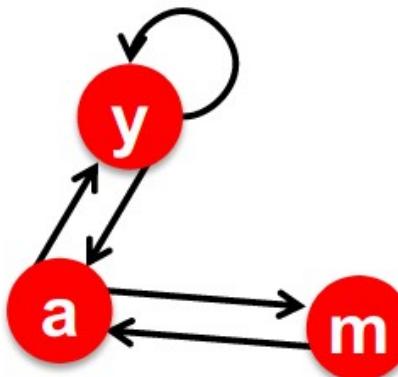
■ Power Iteration:

- Set $r_j \leftarrow 1/N$
- 1: $r'_j \leftarrow \sum_{i \rightarrow j} \frac{r_i}{d_i}$
- 2: $r \leftarrow r'$
- If $|r - r'| > \varepsilon$: goto 1

■ Example:

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

Iteration 0, 1, 2, ...



	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	0
a	$\frac{1}{2}$	0	1
m	0	$\frac{1}{2}$	0

$$r_y = r_y/2 + r_a/2$$

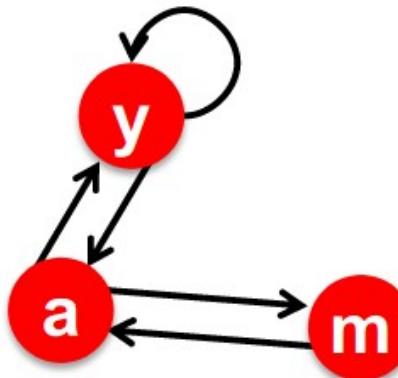
$$r_a = r_y/2 + r_m$$

$$r_m = r_a/2$$

PageRank: How to Solve?

■ Power Iteration:

- Set $r_j \leftarrow 1/N$
- 1: $r'_j \leftarrow \sum_{i \rightarrow j} \frac{r_i}{d_i}$
- 2: $r \leftarrow r'$
- If $|r - r'| > \varepsilon$: goto 1



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

$$\mathbf{r}_y = \mathbf{r}_y/2 + \mathbf{r}_a/2$$

$$\mathbf{r}_a = \mathbf{r}_y/2 + \mathbf{r}_m$$

$$\mathbf{r}_m = \mathbf{r}_a/2$$

■ Example:

$$\begin{bmatrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{bmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \dots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & 3/15 \end{matrix}$$

Iteration 0, 1, 2, ...

PageRank: 3 Questions

$$r_j^{(t+1)} = \sum_{i \rightarrow j} \frac{r_i^{(t)}}{d_i}$$

or
equivalently

$$r = Mr$$

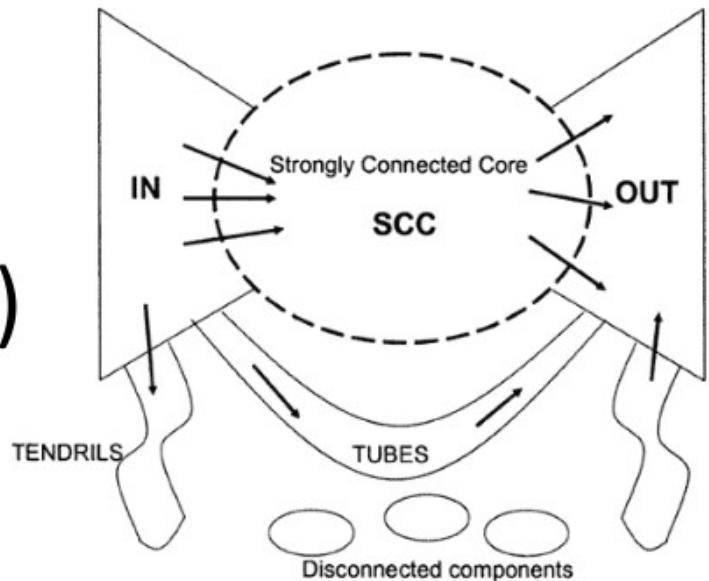
- Does this converge?
- Does it converge to what we want?
- Are the results reasonable?

PageRank: Problems

Two problems:

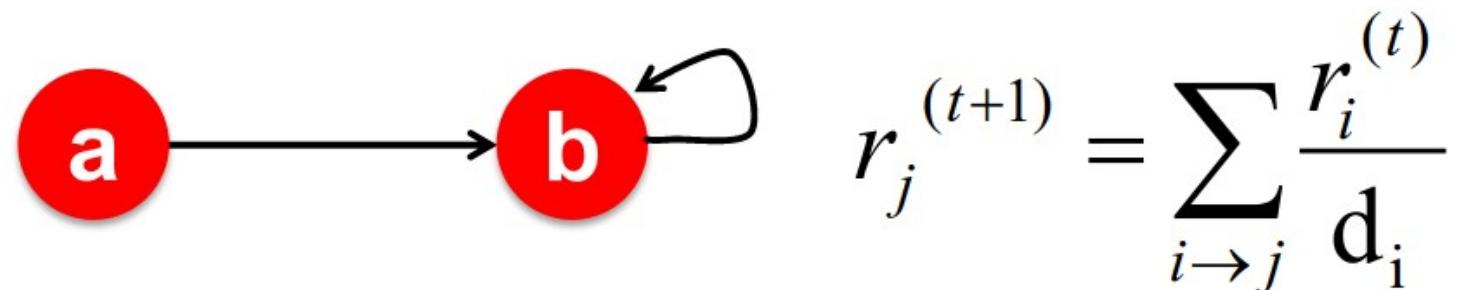
- (1) Some pages are **dead ends** (have no out-links)
 - Such pages cause importance to “leak out”

- (2) **Spider traps**
(all out-links are within the group)
 - Eventually spider traps absorb all importance



Does it converge to what we want?

■ The “Spider trap” problem:



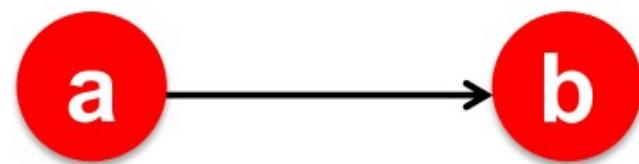
■ Example:

Iteration: 0, 1, 2, 3...

r_a	=	1		0		0		0
r_b		0		1		1		1

Does it converge to what we want?

■ The “Dead end” problem:



$$r_j^{(t+1)} = \sum_{i \rightarrow j} \frac{r_i^{(t)}}{d_i}$$

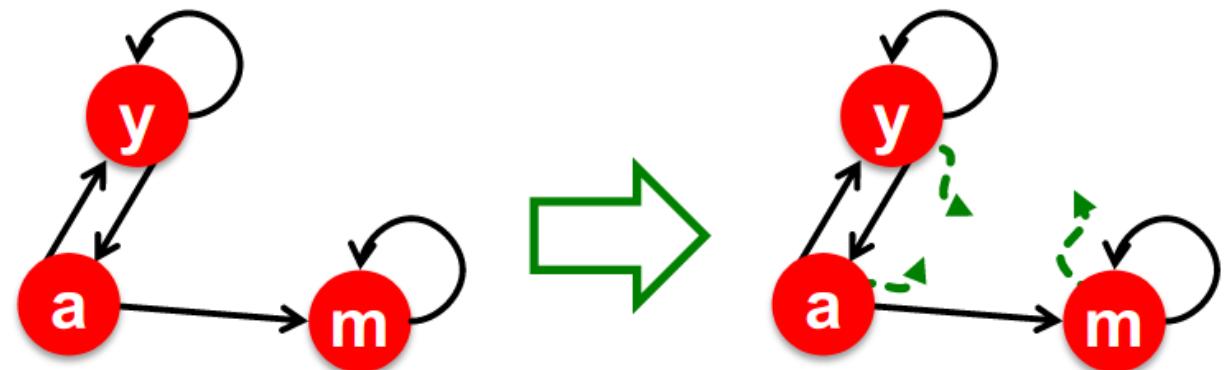
■ Example:

Iteration: 0, 1, 2, 3...

r_a	=	1		0		0		0
r_b		0		1		0		0

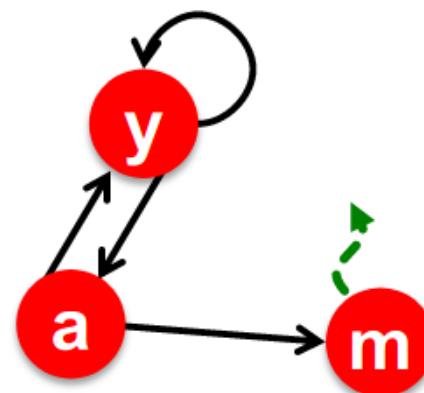
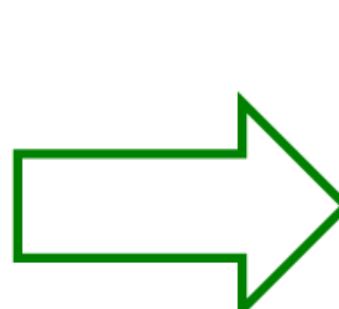
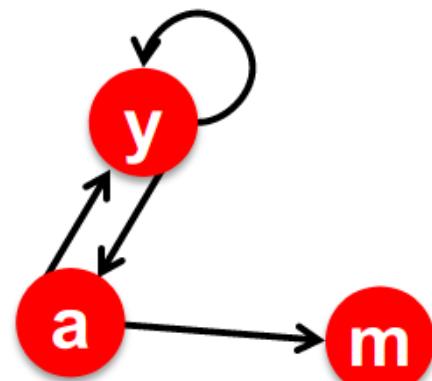
Solution to Spider Traps

- The Google solution for spider traps: At each time step, the random surfer has two options
 - With prob. β , follow a link at random
 - With prob. $1-\beta$, jump to a random page
 - Common values for β are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps



Solution to Dead Ends

- **Teleports:** Follow random teleport links with probability **1.0** from dead-ends
 - Adjust matrix accordingly



	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	0
a	$\frac{1}{2}$	0	0
m	0	$\frac{1}{2}$	0

	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$
a	$\frac{1}{2}$	0	$\frac{1}{3}$
m	0	$\frac{1}{2}$	$\frac{1}{3}$

Final PageRank Equation

- **Google's solution:** At each step, random surfer has two options:
 - With probability β , follow a link at random
 - With probability $1-\beta$, jump to some random page
- **PageRank equation** [Brin-Page, '98]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{n}$$

d_i ... out-degree
of node i

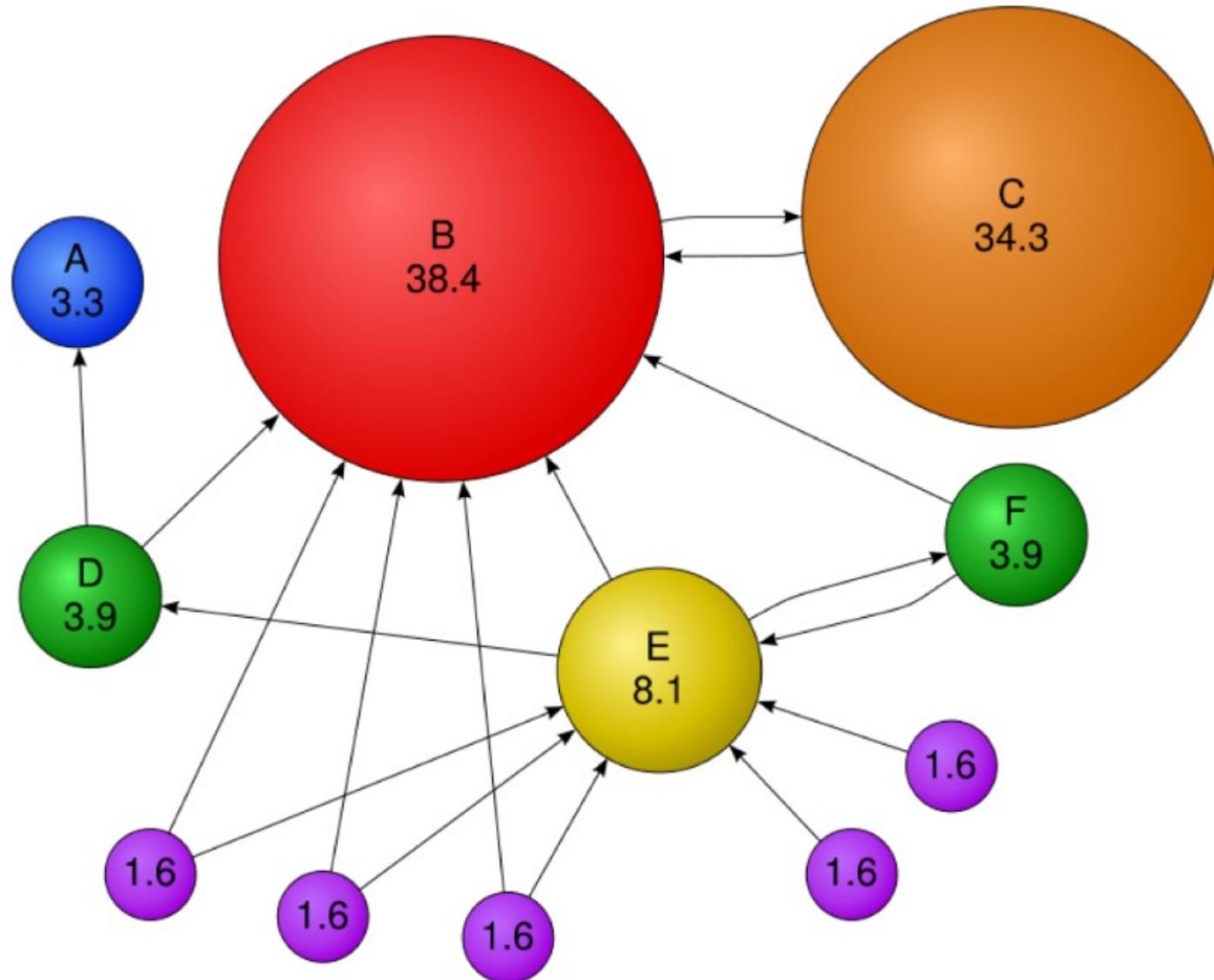
The above formulation assumes that M has no dead ends. We can either preprocess matrix M (**bad!**) or explicitly follow random teleport links with probability 1.0 from dead-ends. See P. Berkhin, *A Survey on PageRank Computing*, Internet Mathematics, 2005.

The PageRank Algorithm

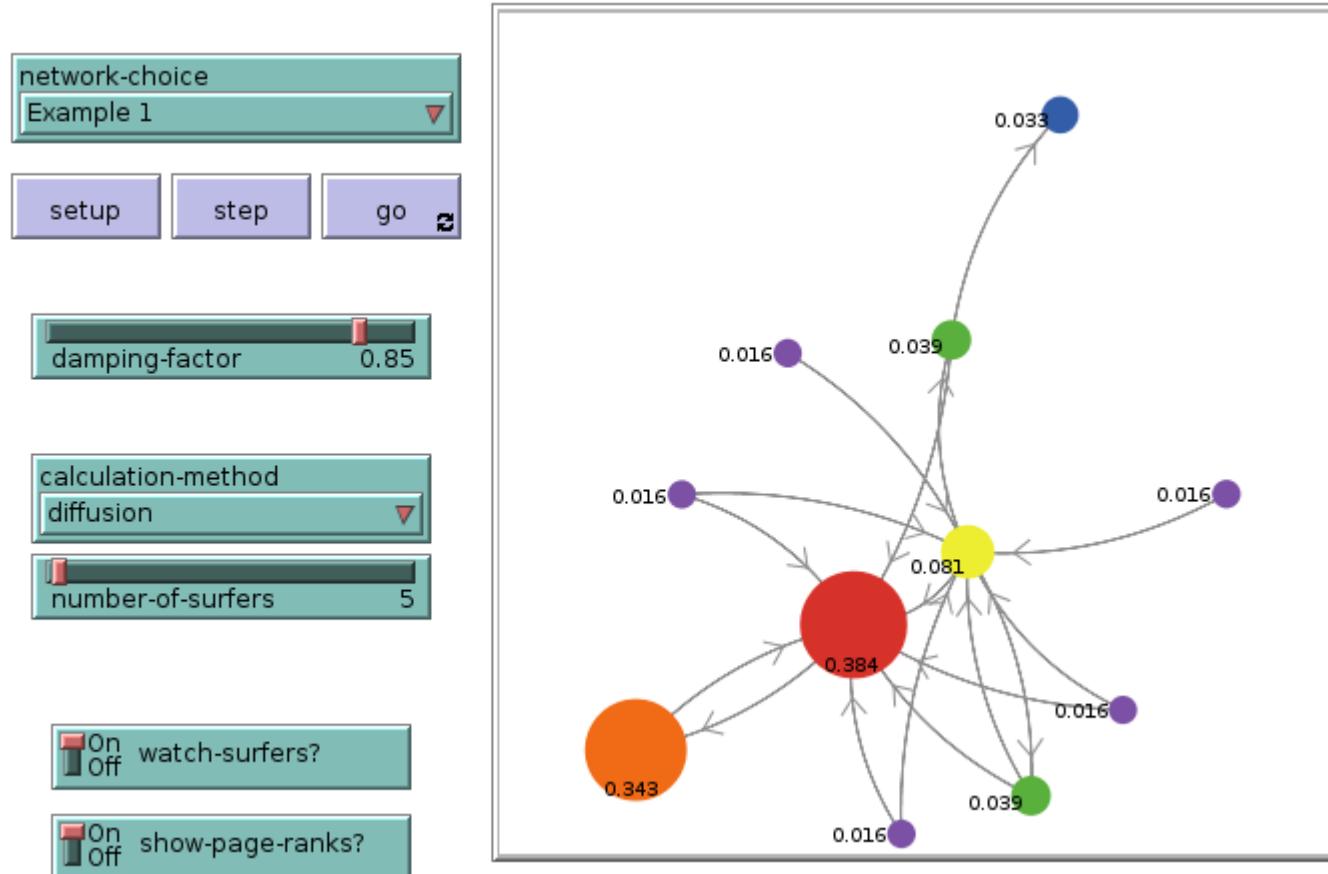
- **Input:** Graph G and parameter β
 - Directed graph G with spider traps and dead ends
 - Parameter β
- **Output:** PageRank vector r
 - Set: $r_j^{(0)} = \frac{1}{N}, t = 1$
 - do:
 - $\forall j: r'_j^{(t)} = \sum_{i \rightarrow j} \beta \frac{r_i^{(t-1)}}{d_i}$
 - $r'_j^{(t)} = \mathbf{0}$ if in-deg. of j is 0
 - Now re-insert the leaked PageRank:
 $\forall j: r_j^{(t)} = r'_j^{(t)} + \frac{1-s}{N}$ where: $s = \sum_j r'_j^{(t)}$
 - $t = t + 1$
 - while $\sum_j |r_j^{(t)} - r_j^{(t-1)}| > \varepsilon$

Example

Node size proportional to the PageRank score



NetLogo: PageRank

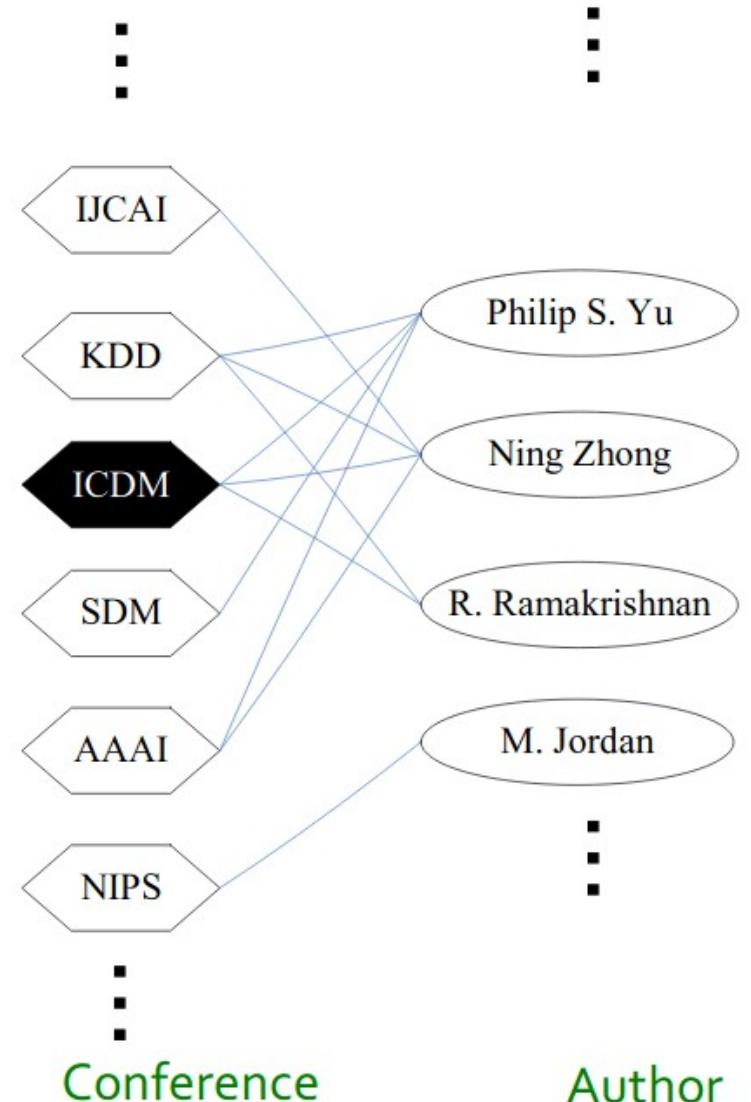


PageRank.nlogo

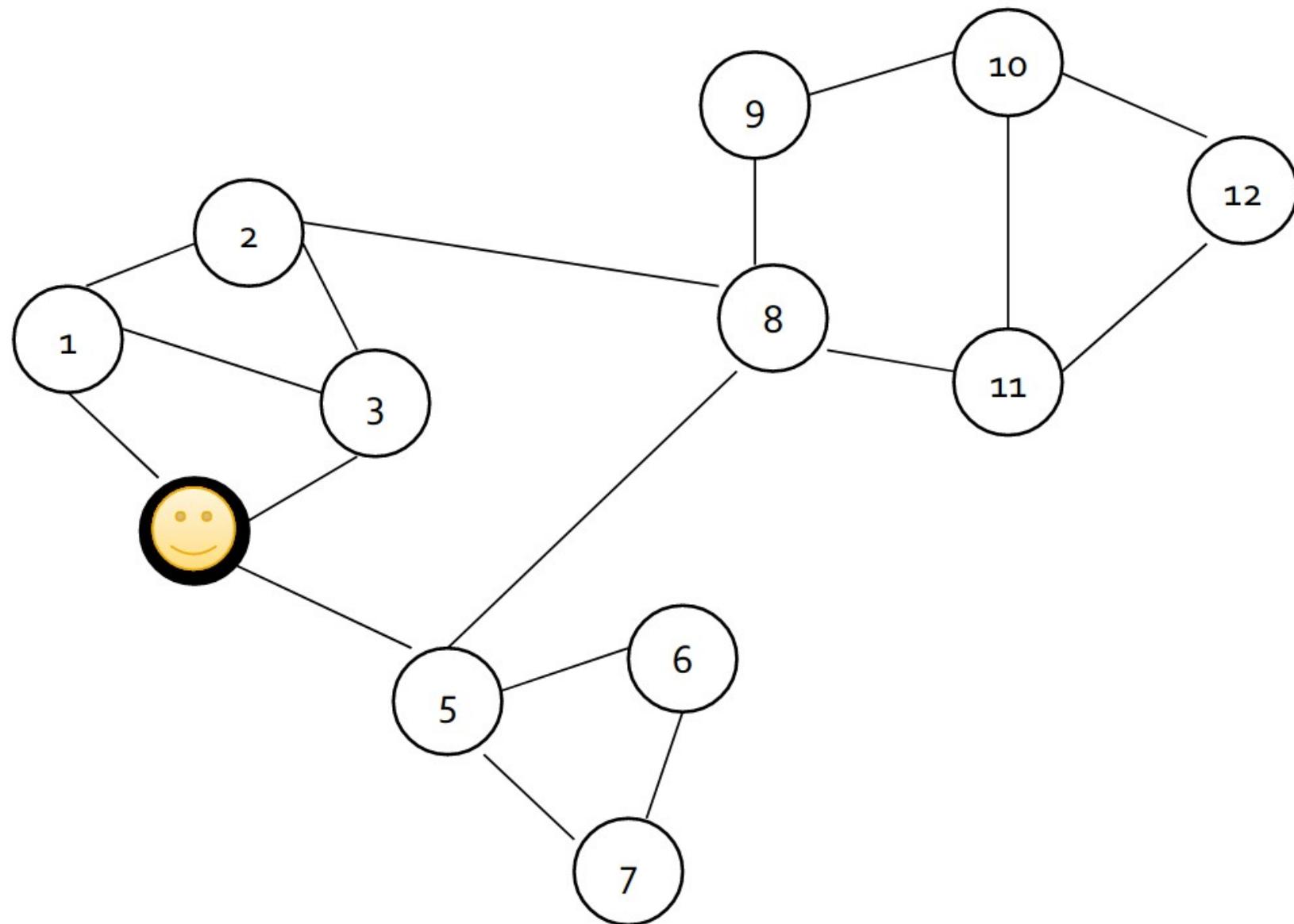
Random Walk Restarts and Personalized PageRank

Example Application: Graph Search

- **Given:**
Conferences-to-authors
graph
 - **Goal:**
Proximity on graphs
 - Q: What is most related
conference to ICDM?



Random Walk with Restarts



Personalized PageRank

- **Goal:** Evaluate pages not just by popularity but by how close they are to the topic
- **Teleporting can go to:**
 - Any page with equal probability
 - PageRank (we used this so far)
 - A topic-specific set of “relevant” pages
 - Topic-specific (personalized) PageRank (S ...teleport set)
$$\begin{aligned} M'_{ij} &= \beta M_{ij} + (1 - \beta)/|S| && \text{if } i \in S \\ &= \beta M_{ij} && \text{otherwise} \end{aligned}$$
 - A single page/node ($|S| = 1$),
 - Random Walk with Restarts

PageRank: Applications

■ Graphs and web search:

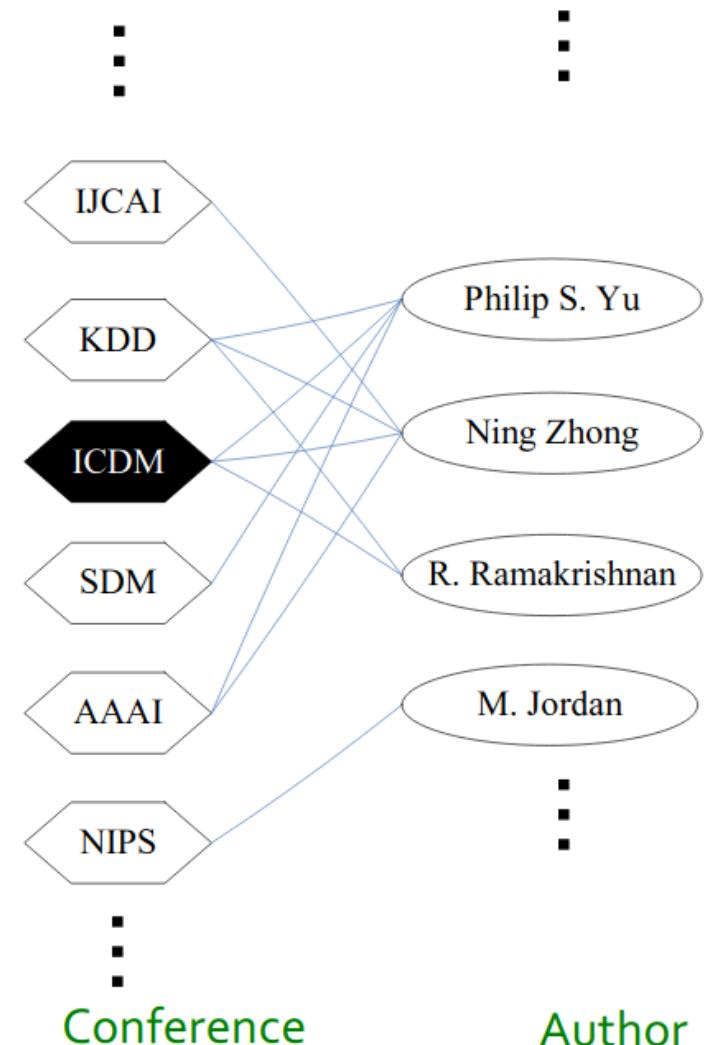
- Ranks nodes by “importance”

■ Personalized PageRank:

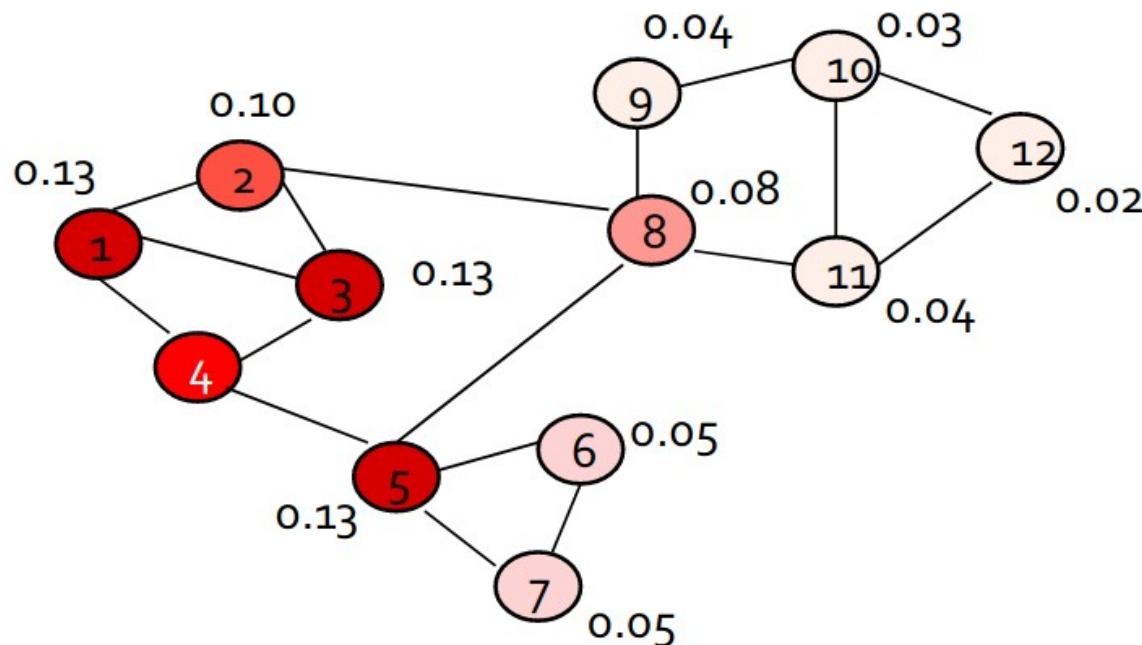
- Ranks proximity of nodes to the teleport set S

■ Proximity on graphs:

- **Q:** What is most related conference to ICDM?
- **Random Walks with Restarts**
 - Teleport back to the starting node:
 $S = \{ \text{single node} \}$



Random Walk with Restarts



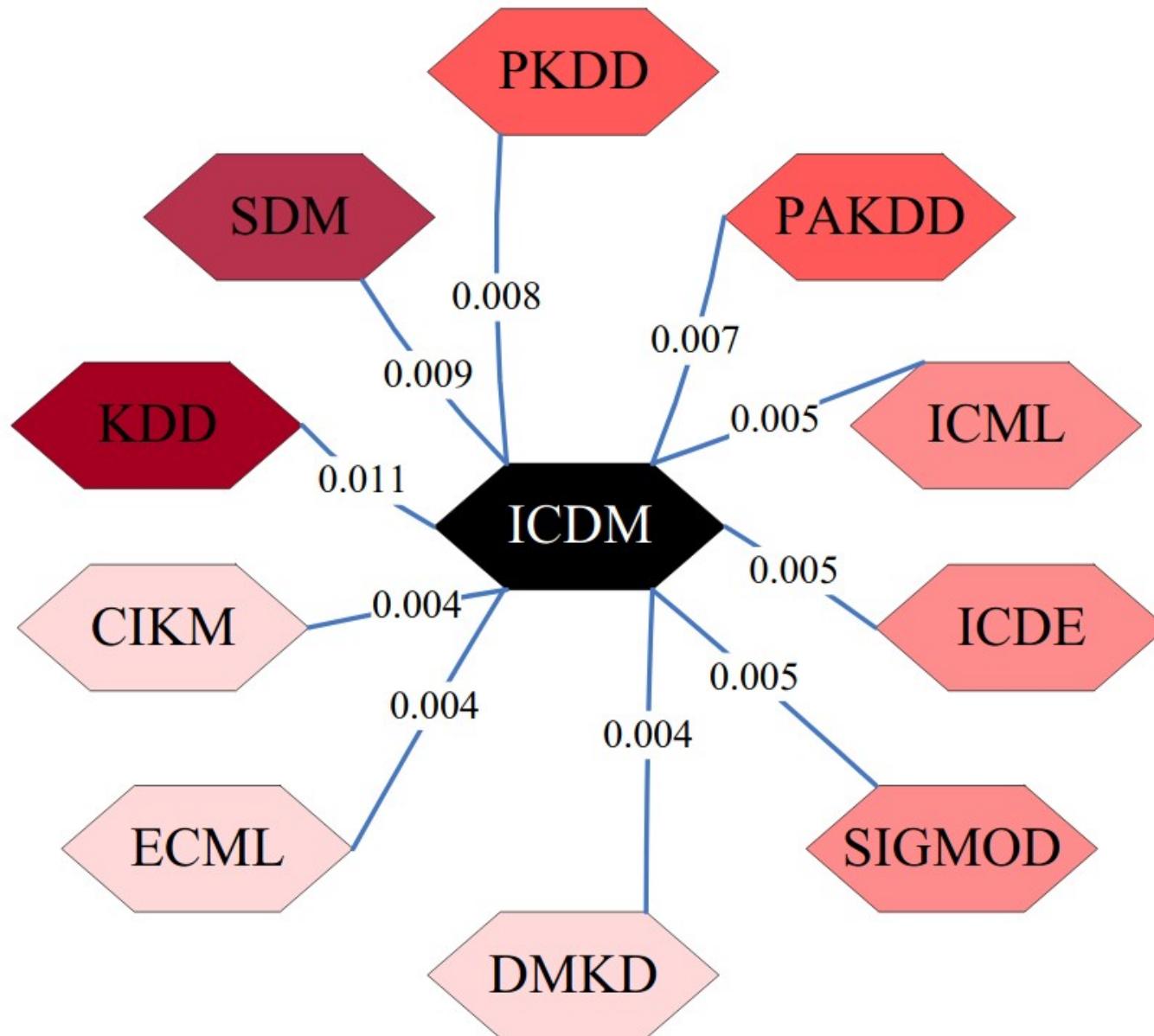
	Node 4
Node 1	0.13
Node 2	0.10
Node 3	0.13
Node 4	/
Node 5	0.13
Node 6	0.05
Node 7	0.05
Node 8	0.08
Node 9	0.04
Node 10	0.03
Node 11	0.04
Node 12	0.02

$S=\{4\}$

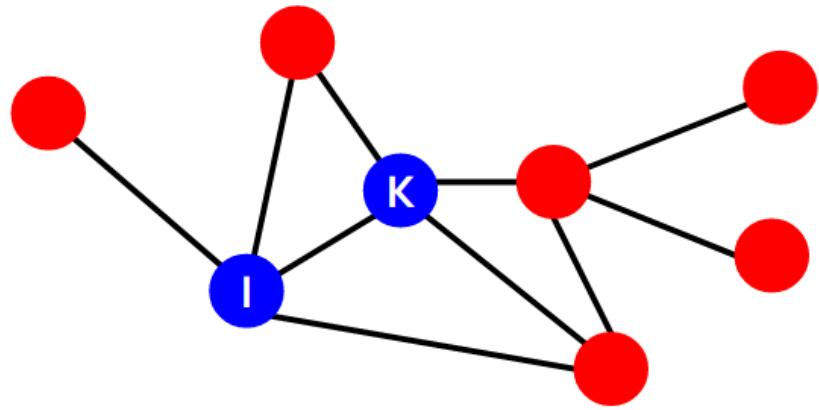
Notice: Nearby nodes have higher scores (are more red)

Ranking vector

Most Related Conferences to ICDM



Personalized PageRank



Graph of CS conferences

Q: Which conferences
are closest to KDD &
ICDM?

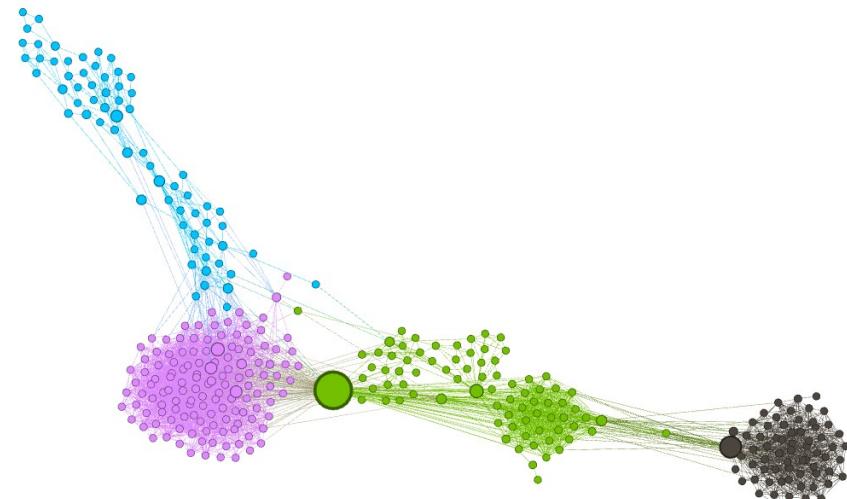
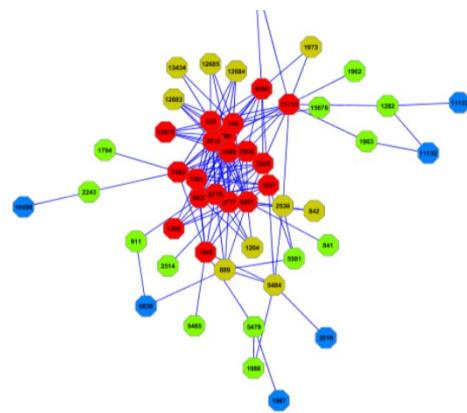
A: Personalized
PageRank with
teleport set $S=\{KDD,$
 $ICDM\}$

Community Structure in Networks



Pedro Ribeiro

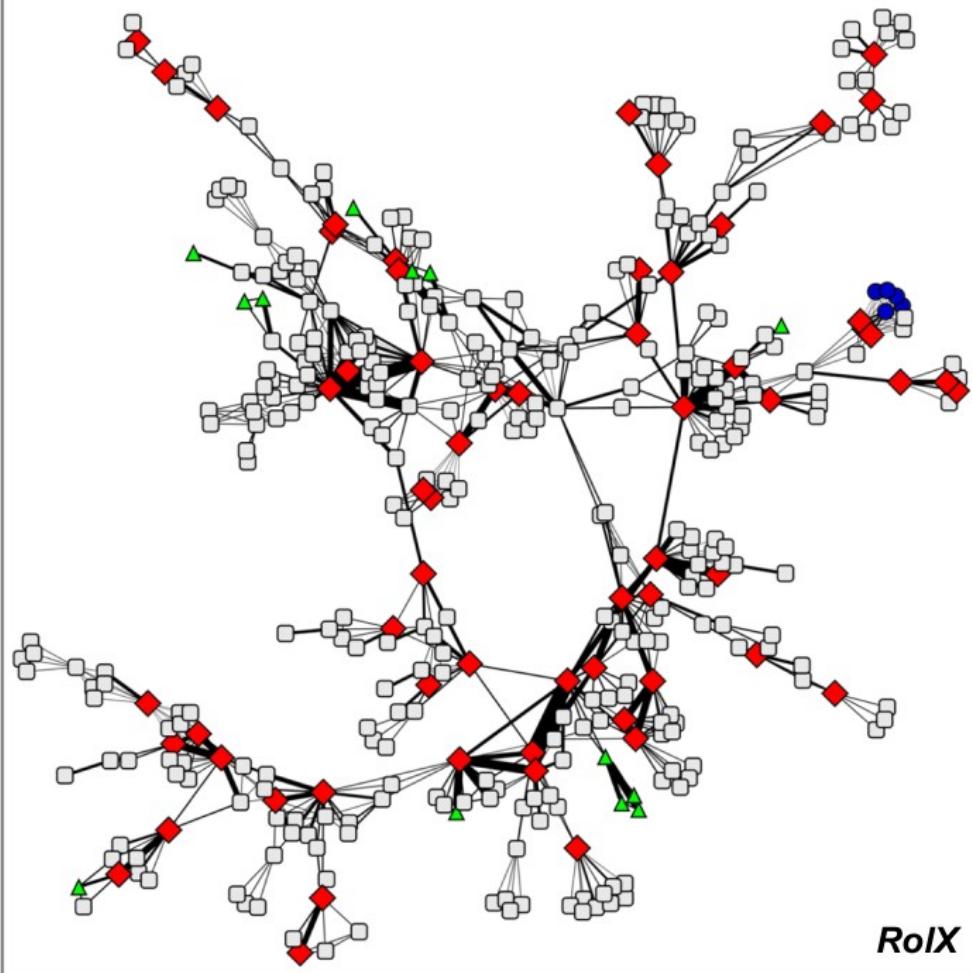
(DCC/FCUP & CRACS/INESC-TEC)



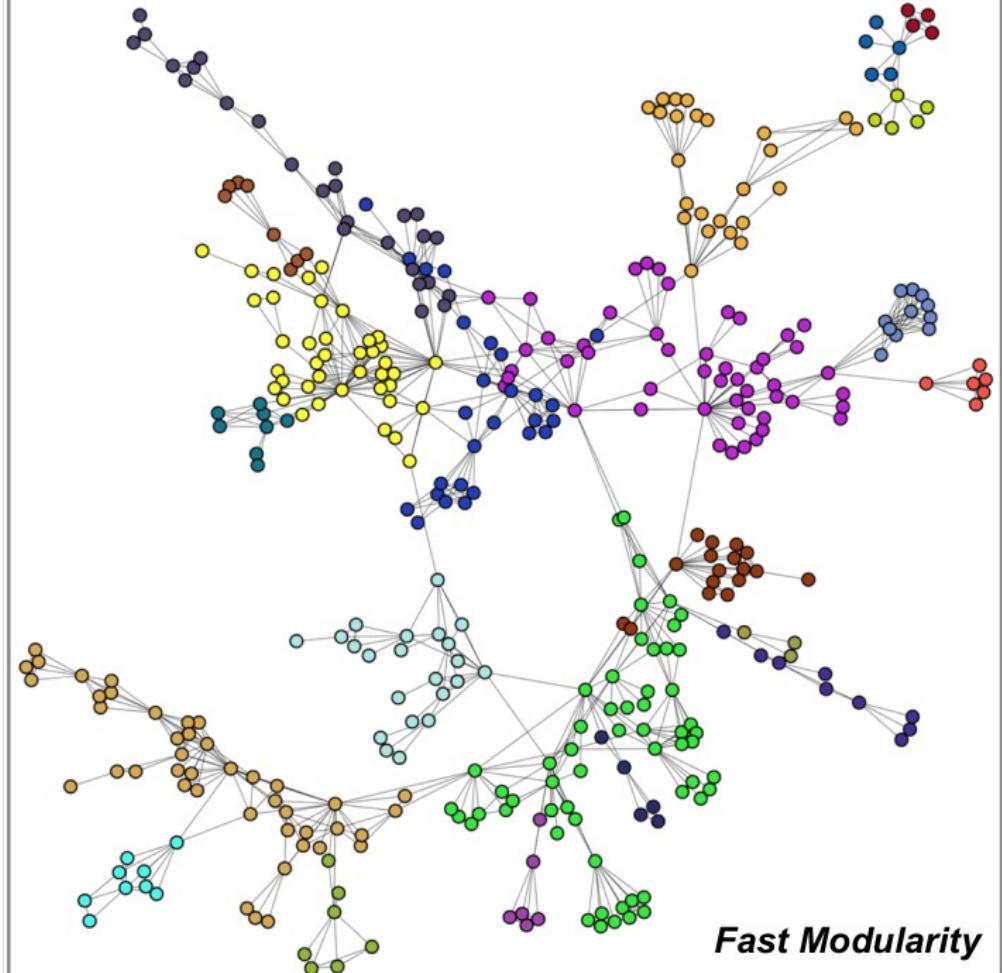
(Mainly selected slides from Jure Leskovec and Gonzalo Mateos)

Roles and Communities

Roles



Communities



Henderson, et al., KDD 2012

Nodes with different structural roles
(connector node, bridge node, etc.)

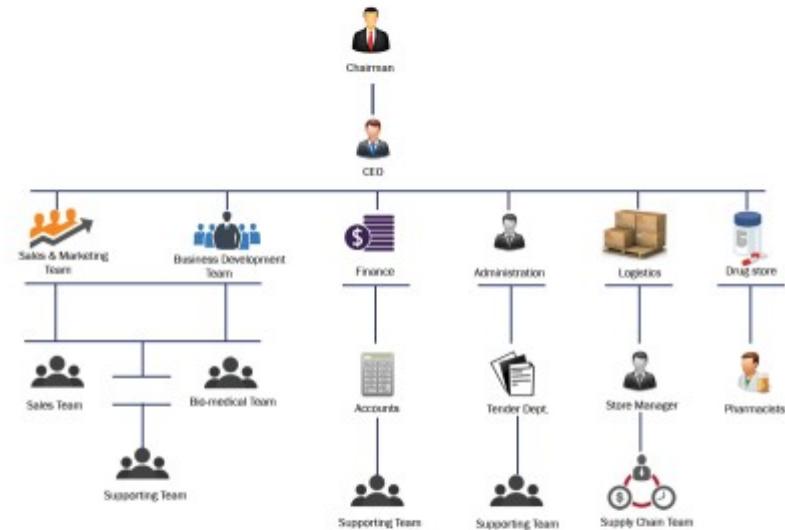
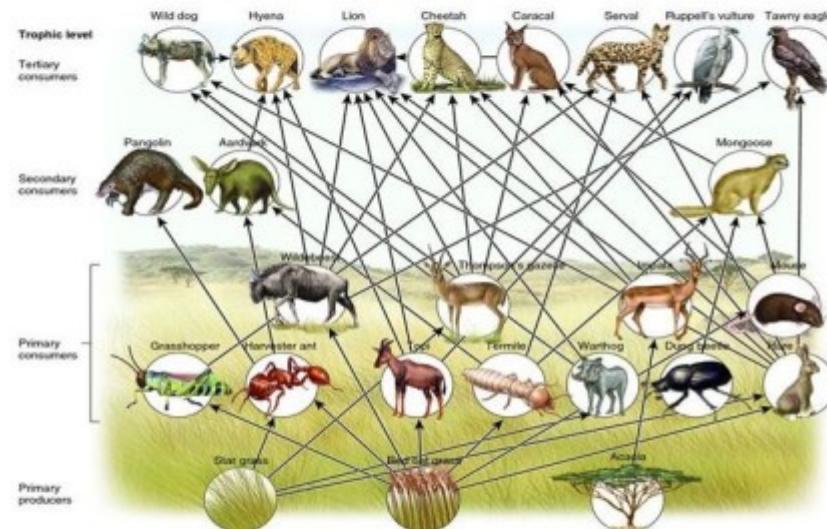
Clauset, et al., Phys. Rev. E 2004

Nodes belonging to the same
cluster/community

Structural Roles

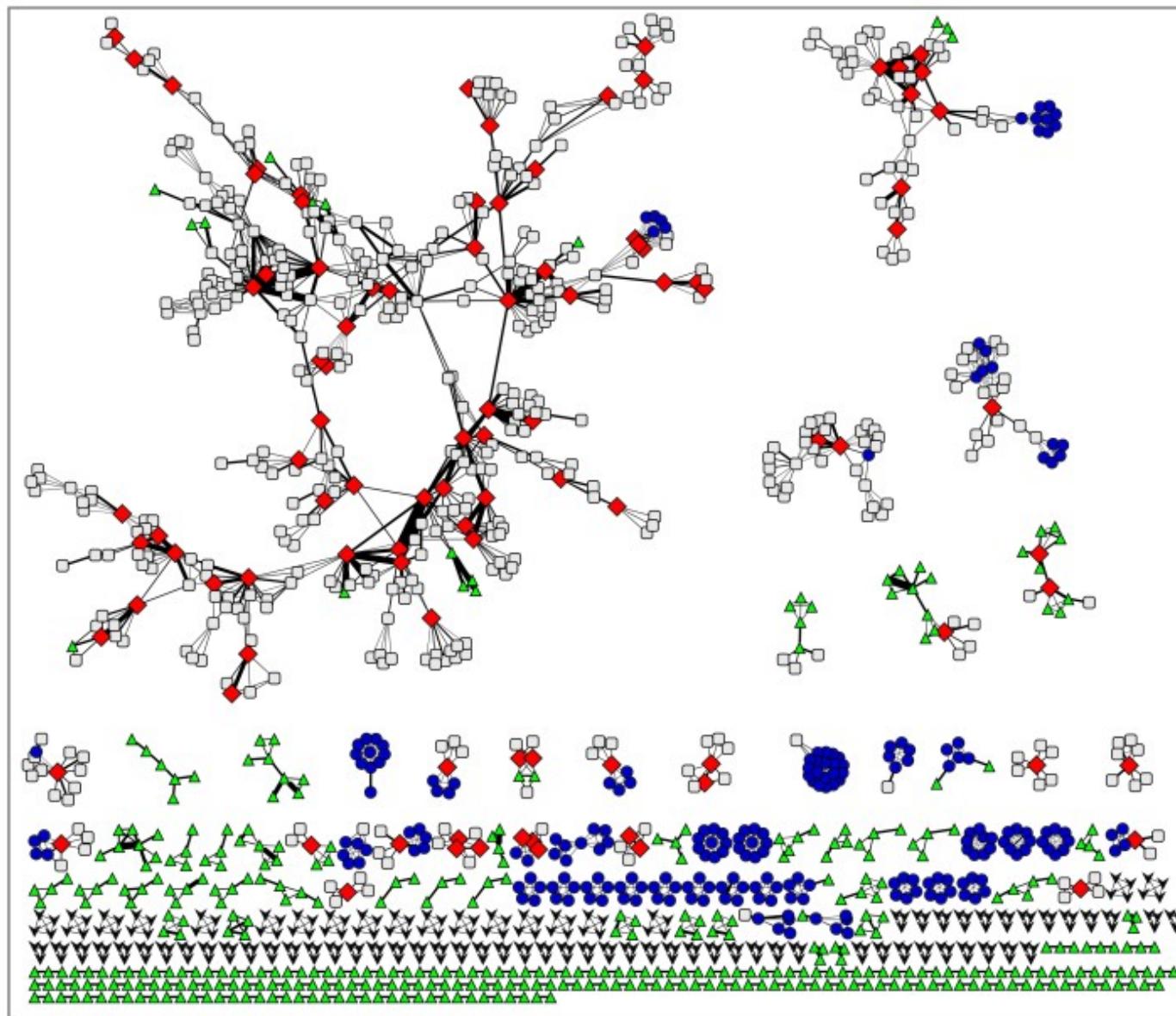
What are Roles?

- Roles are “functions” of nodes in a network:
 - Roles of species in **ecosystems**
 - Roles of individuals in **companies**



- Roles are measured by structural behaviors:
 - Centers of stars
 - Members of cliques
 - Peripheral nodes, etc.

Examples of Roles



- ◆ centers of stars
- members of cliques
- ▲ peripheral nodes

Network Science
Co-authorship network
[Newman 2006]

Roles vs Groups in Networks

- **Role:** A collection of nodes which have similar positions in a network:
 - Roles are based on the similarity of ties among subsets of nodes
 - Different from **community** (or cohesive subgroup)
 - Group is formed based on adjacency, proximity or reachability
 - This is typically adopted in current data mining

Nodes with the same role need not be in direct, or even indirect interaction with each other

Roles and Communities

- **Roles:**

- A group of nodes with similar structural properties

- **Communities:**

- A group of nodes that are well-connected to each other

- Roles and communities **are complementary**

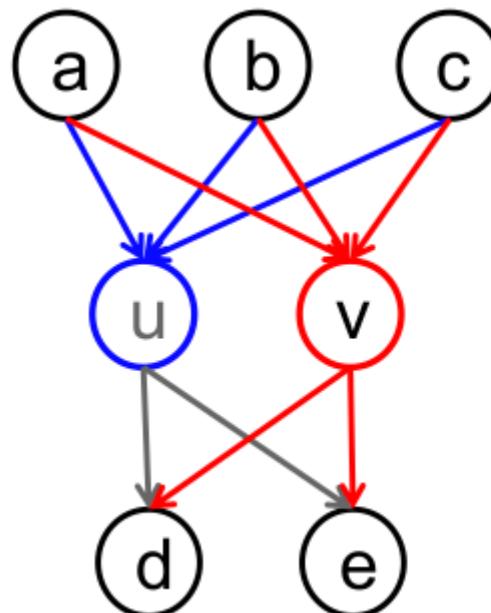
- Consider the social network of a CS Dept:

- **Roles:** Faculty, Staff, Students

- **Communities:** AI Lab, Info Lab, Theory Lab

Roles: More Formally

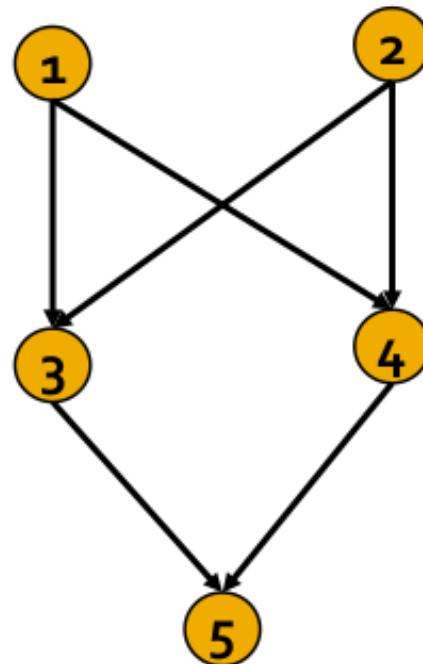
- **Structural equivalence:** Nodes u and v are structurally equivalent if they have the same relationships **to all other nodes** [Lorrain & White 1971]
 - Structurally equivalent nodes are likely to be similar in other ways – *i.e.*, friendships in social networks



Structural Equivalence

- Nodes u and v are **structurally equivalent**:
 - For all the other nodes k , node u has tie to k iff node v has tie to k

- Example:



Adjacency matrix

	1	2	3	4	5
1	-	0	1	1	0
2	0	-	1	1	0
3	0	0	-	0	1
4	0	0	0	-	1
5	0	0	0	0	-

- E.g., nodes 3 and 4 are structurally equivalent

Discovering Structural Roles

Why are roles important?

Task	Example Application
Role query	Identify individuals with similar behavior to a known target
Role outliers	Identify individuals with unusual behavior
Role dynamics	Identify unusual changes in behavior
Identity resolution	Identify/de-anonymize, individuals in a new network
Role transfer	Use knowledge of one network to make predictions in another
Network comparison	Compute similarity of networks, determine compatibility for knowledge transfer

War Story

Evolutionary Role Mining in Complex Networks by Ensemble Clustering

Sarvenaz Choobdar, Pedro Ribeiro, Fernando Silva

CRACS & INESC-TEC

DCC-FCUP, Universidade do Porto, Portugal

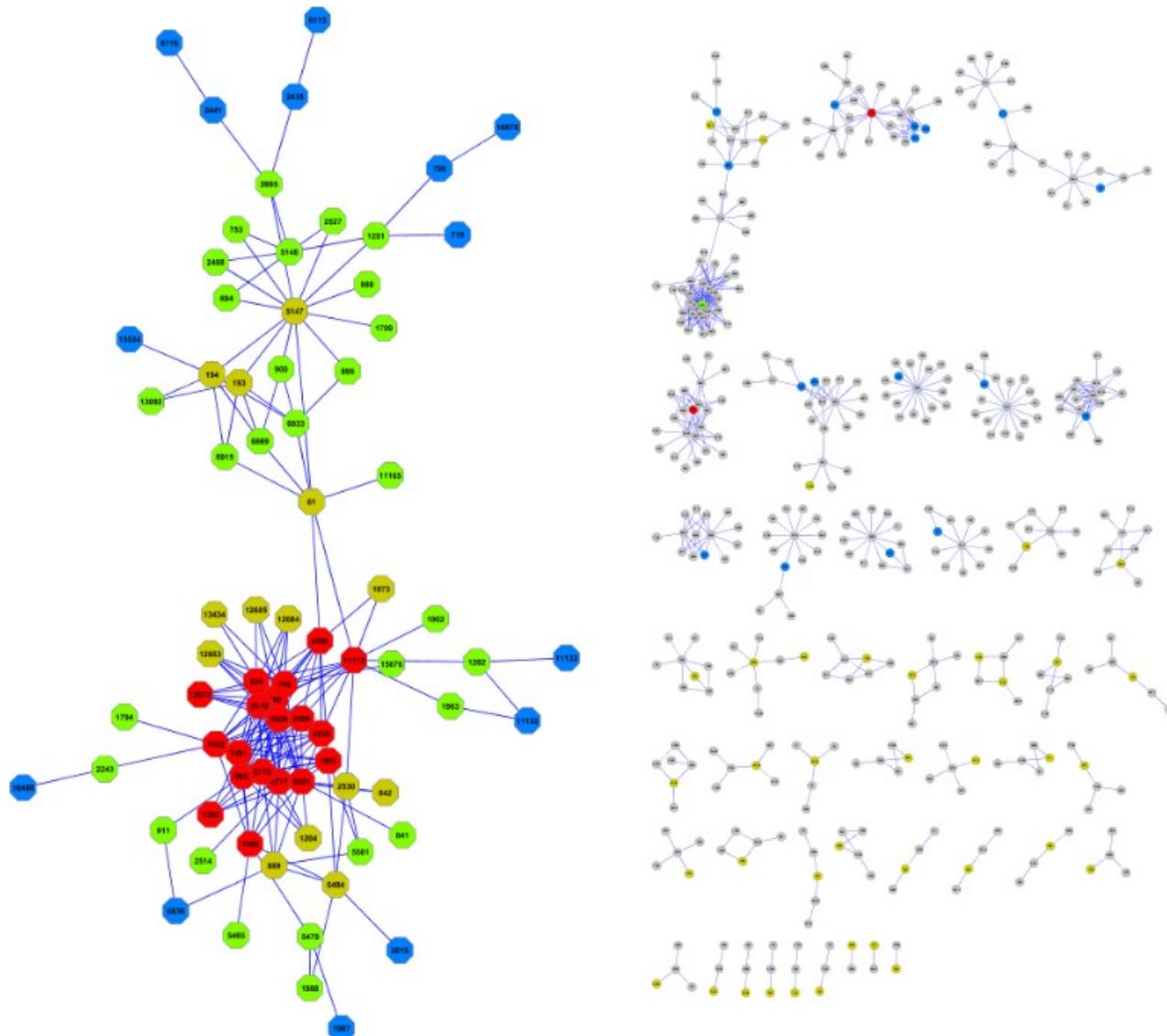
{sarvenaz,pribeiro,fds}@dcc.fc.up.pt

- the normalized node degree: quantifies the linkage of node i ; it is the degree of node i divided by the sum of all nodes' degree in the network.
- the normalized average degree: shows the intensity of connectivity in the neighborhood of node i ; it is calculated by averaging over all degree of immediate neighbors of node i .
- the coefficient variation of the degrees of the immediate neighbors of a node (cv): characterizes the coherence of the connectivity; it is the standard deviation of the degrees in the neighborhood of node i .
- the clustering coefficient: quantifies the connectivity between neighbors; it is measured as the proportion of existing connections between neighbors of node i to the number of all possible links between them [25].
- the locality index: characterizes the structure of neighbors' connectivity to rest of the network; it is the ratio of links within the neighborhood to the number of links to the nodes outside of neighborhood.

Algorithm 1 Evolutionary Role Mining (ERM)

```
1: procedure ERM( $G_T, K, wFun(C), clustAlgo(M, K)$ )
2:   for  $t$  in  $1 : T$  do
3:      $X_t \leftarrow localProperties(G_t)$ 
4:      $C_t \leftarrow kmeans(X_t, K)$ 
5:      $C \leftarrow C \cup C_t$ 
6:   end for
7:    $\{\alpha_1, \dots, \alpha_T\} \leftarrow wFun(C)$ 
8:   for  $t$  in  $1 : T$  do
9:      $M \leftarrow M + pairwiseSimilarity(G, C_t) * \alpha_t$ 
10:  end for
11:   $C_T \leftarrow clustAlgo(M, K)$ 
12:  return  $C_T$ 
13: end procedure
```

War Story



(b) Color-code by role of nodes, identified by proposed method

War Story

Pairwise structural role mining for user categorization in information cascades

Sarvenaz Choobdar, Pedro Ribeiro, Fernando Silva

CRACS and INESC-TEC

University of Porto, Portugal

Email: {sarvenaz,prebeiro,fds}@dcc.fc.up.pt

Abstract—The tendency of users to connect with peers of similar interests and social demography (homophily) is one of the sources of information for user behavior modeling and classification. However this is yet an open question for structural roles where nodes at similar structural position in the network play the same roles: are structurally equivalent nodes more prone to have connections between themselves? In this paper, we tackle this open question by studying the patterns of homophily for structural roles. We propose a new method named SR-Diffuse to simultaneously identify structural roles in a network and to model the role membership matrix of users. In this method, we integrate pairwise role dependency alongside with structural features of users for role mining. We show that pairwise role dependency is necessary to distinguish some structural roles but it is a misleading factor for some others. We design an optimization model to capture structural roles with the guidance of pairwise dependency, and devise an iterative algorithm to learn structural roles simultaneously from structural properties and social dependency of users. We examine the efficacy of our new method in a users classification problem for information cascades. We compare the predictability of discovered roles by our method against some baseline methods for predicting social classes of users in different information cascades in two social networks, Flickr and Digg. The experimental results suggest that our method can improve the quality of roles membership of users and can better represent the profile of users in the network, hence it is a better predictor for social classes of users in an information cascade.

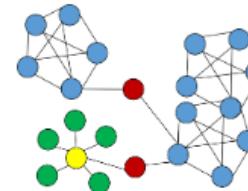
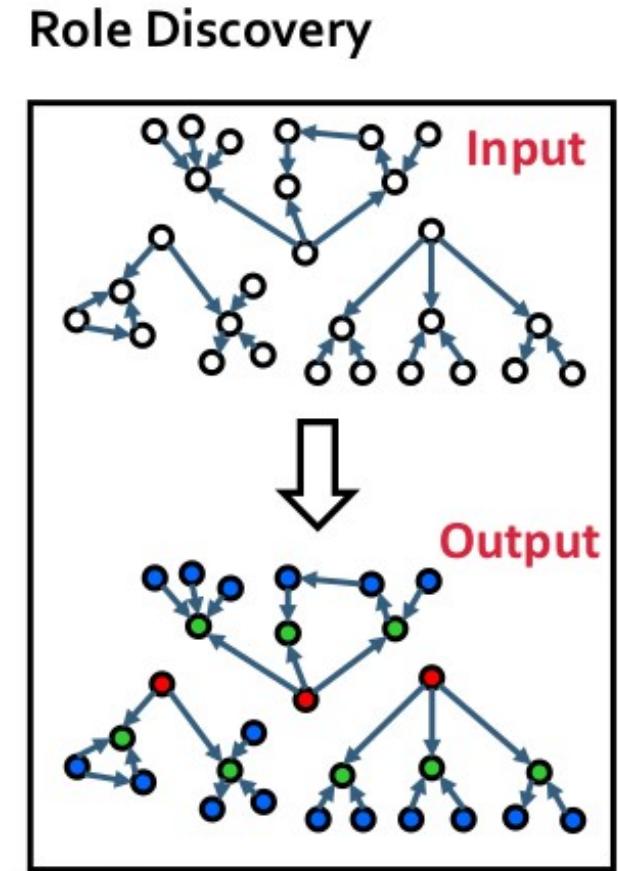


Fig. 1: Pairwise dependency across structural roles, different colors correspond to different structural roles; the pairwise role dependency exists in some structural roles such as member-of-clique (blue nodes) but it does not hold on some others such as member-of-star (green nodes).

structural position may have a tendency to have connections between themselves. Figure 1 exemplifies that, with the blue nodes (member-of-clique) having connections to other blue nodes. However, this is not the case for all types of structural roles. For instance, the green nodes (member-of-star) have no connections to other green nodes, as their structural features do not give origin to pairwise connections. In this paper, one of our main goals is to incorporate pairwise dependency of different structural roles in role mining framework. For that, we first examine how actually the pairwise relations are across

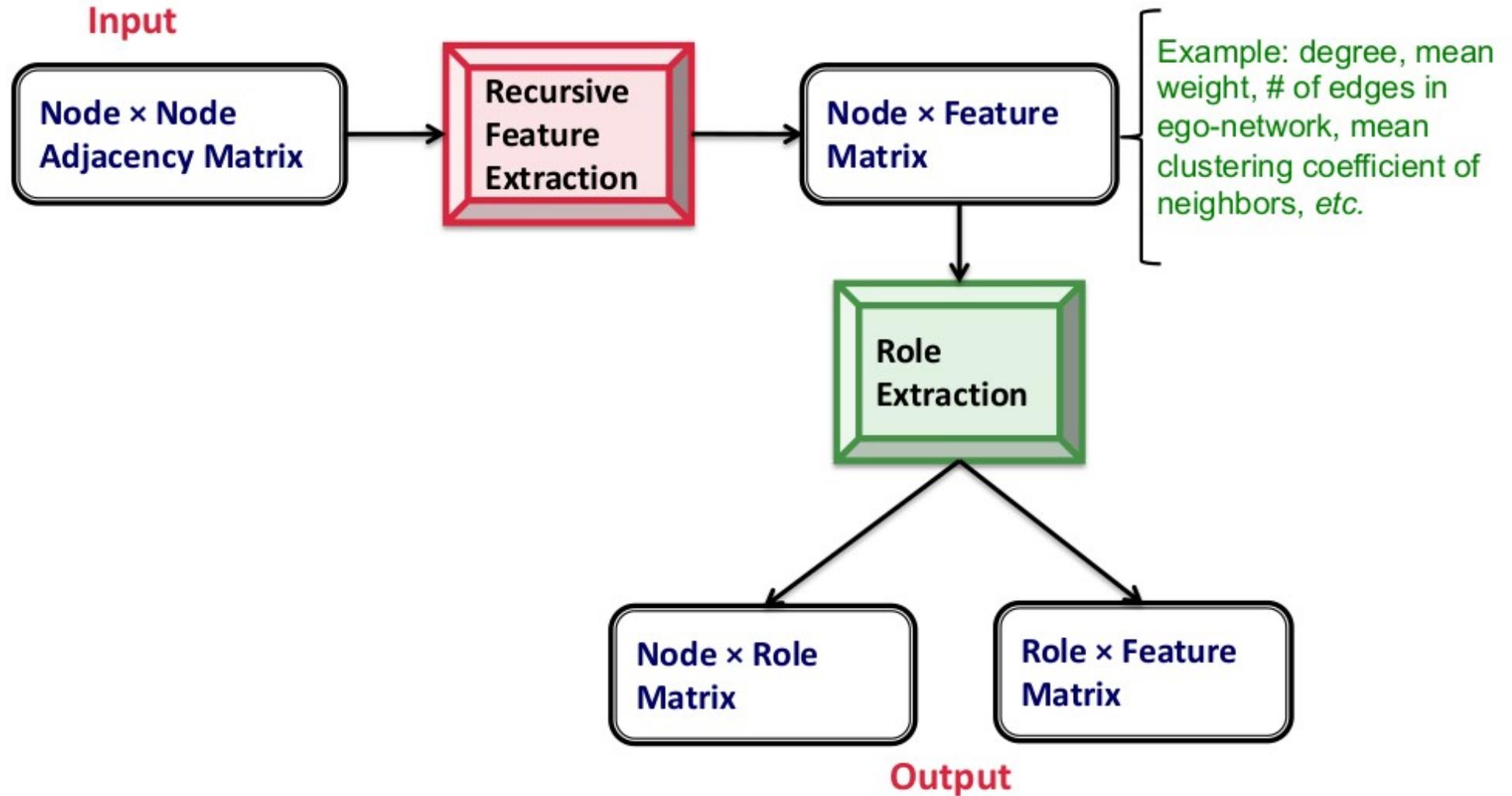
- **RoIX:** Automatic discovery of nodes' structural roles in networks
[Henderson, et al. 2011b]

- Unsupervised learning approach
- No prior knowledge required
- Assigns a mixed-membership of roles to each node
- Scales linearly in #(edges)



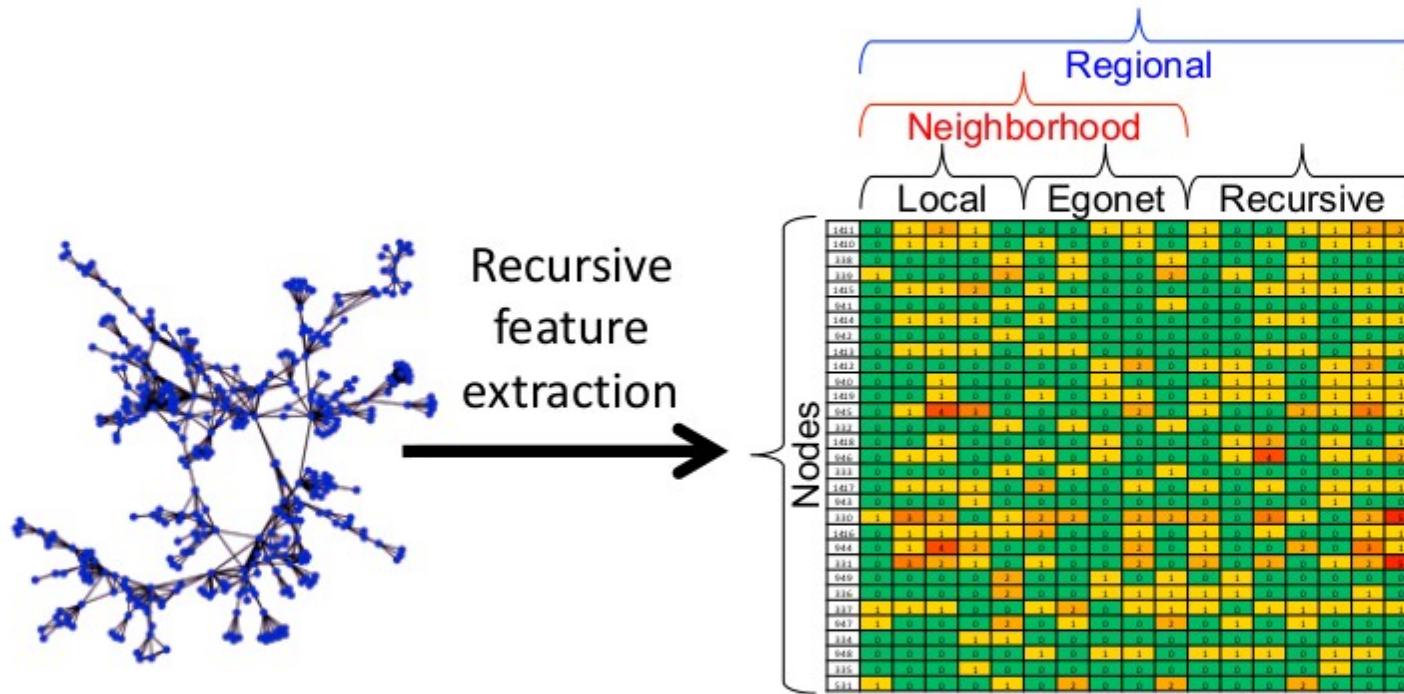
- ✓ Automated discovery
- ✓ Behavioral roles
- ✓ Roles generalize

RoI_X: Approach Overview



RoIX: Recursive Feature Extraction

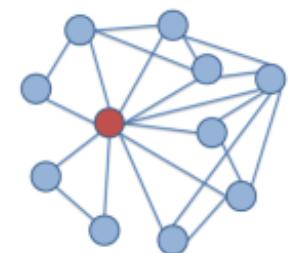
- **Recursive feature extraction** [Henderson, et al. 2011a] turns network connectivity into structural features



- **Neighborhood features:** What is a node's connectivity pattern?
- **Recursive features:** To what kinds of nodes is a node connected?

RoIX: Recursive Feature Extraction

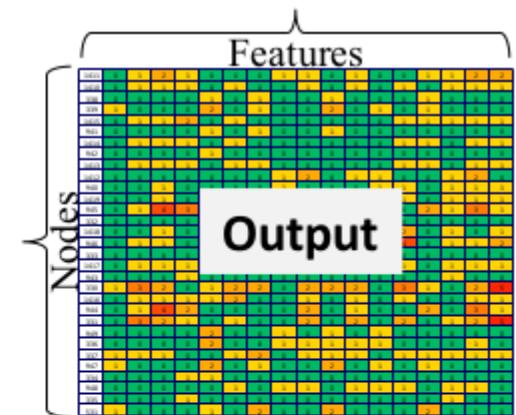
- **Idea:** Aggregate features of a node and use them to **generate new recursive features**
- **Base set of a node's neighborhood features:**
 - **Local features:** All measures of the node degree:
 - If network is directed, include in- and out-degree, total degree
 - If network is weighted, include weighted feature versions
 - **Egonetwork features:** Computed on the node's egonet:
 - **Egonet** includes the node, its neighbors, and any edges in the induced subgraph on these nodes
 - #(within-egonet edges),
#(edges entering/leaving egonet)



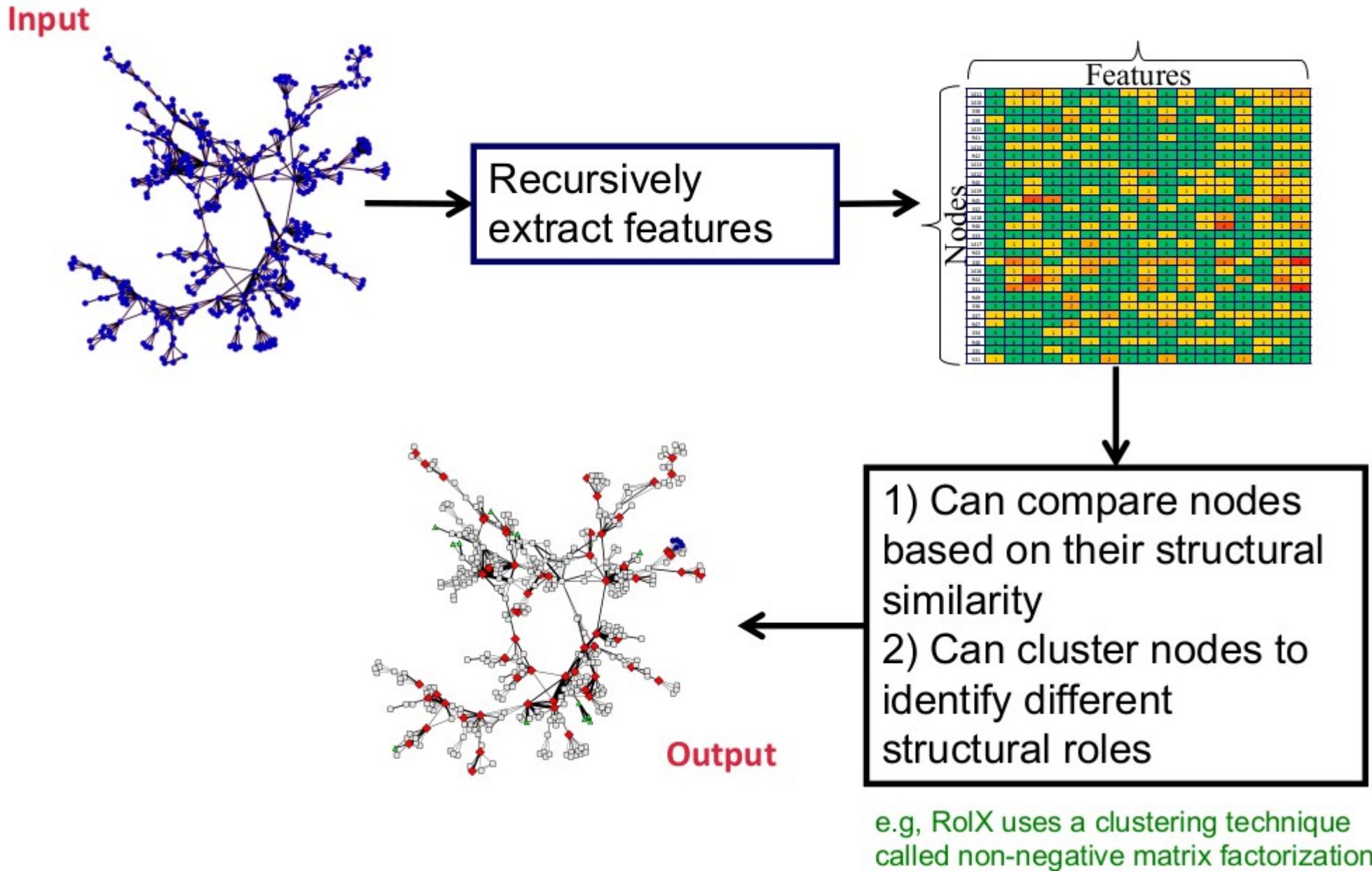
Egonet for **red node**

RoIX: Recursive Feature Extraction

- Start with the base set of node features
- Use the **set of current node features** to generate **additional features**:
 - Two types of **aggregate functions**: means and sums
 - E.g., mean value of “unweighted degree” feature among all neighbors of a node
 - Compute means and sums over all current features, including other recursive features
 - Repeat
- The number of possible recursive features **grows exponentially** with each recursive iteration:
 - Reduce the number of features using a **pruning technique**:
 - Look for pairs of features that are highly correlated
 - Eliminate one of the features whenever two features are correlated above a user-defined threshold



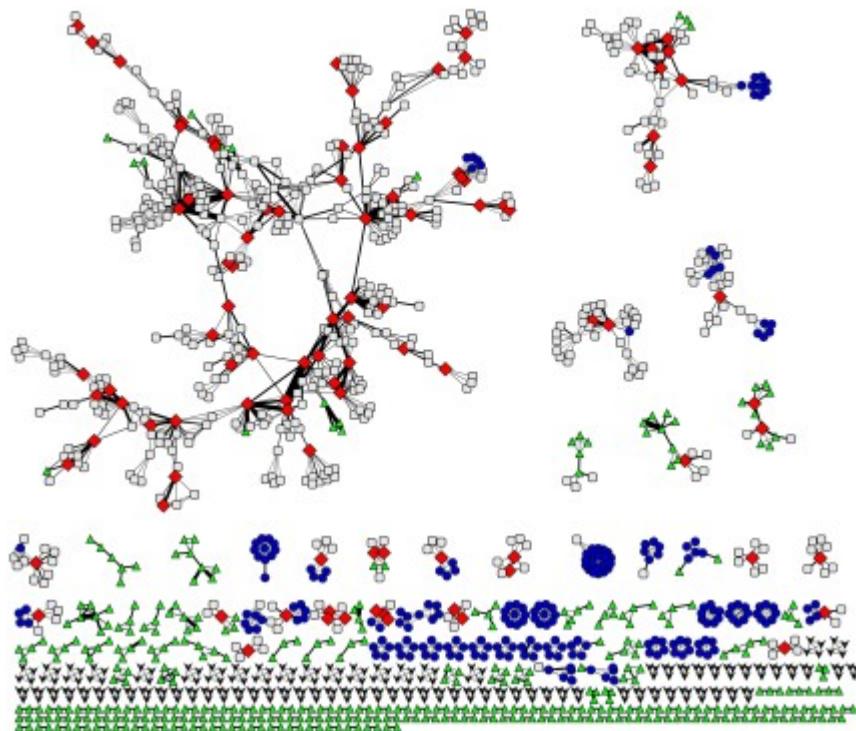
RoIX: Role Extraction



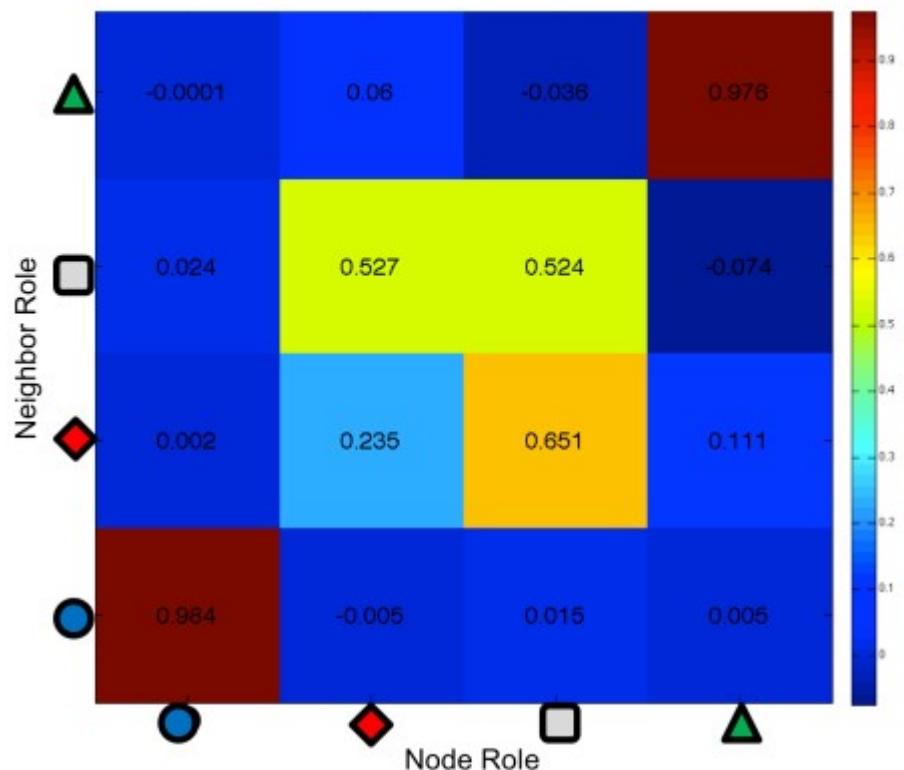
Application: Structural Similarity

- **Task:** Cluster nodes based on their structural similarity
- **Two networks:**
 - Network science co-authorship network:
 - Nodes: Network scientists; Edges: The number of co-authored papers
 - Political books co-purchasing network:
 - Nodes: Political books on Amazon; Edges: Frequent co-purchasing of books by the same buyers
- **Setup:** For each network:
 - Use RolX to assign each node a distribution over the set of discovered, structural roles
 - Determine similarity between nodes by comparing their role distributions

Structural Sim: Co-Authorship



Role-colored graph: each node is colored by the primary role that RolX finds



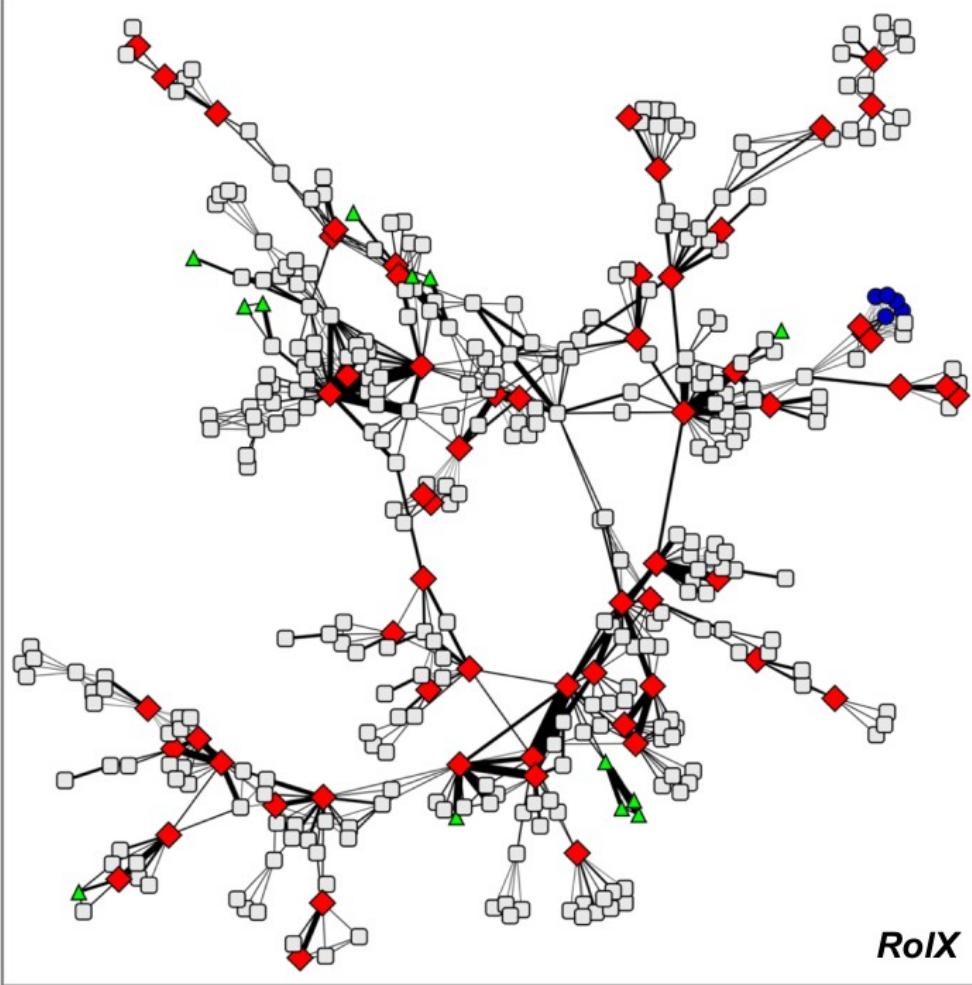
Making sense of roles:

- **Blue circle: Tightly knit**, nodes that participate in tightly-coupled groups
- **Red diamond: Bridge nodes**, that connect groups of nodes
- **Gray rectangle: Main-stream**, most of nodes, neither a clique, nor a chain
- **Green triangle: Pathy**, nodes that belong to elongated clusters

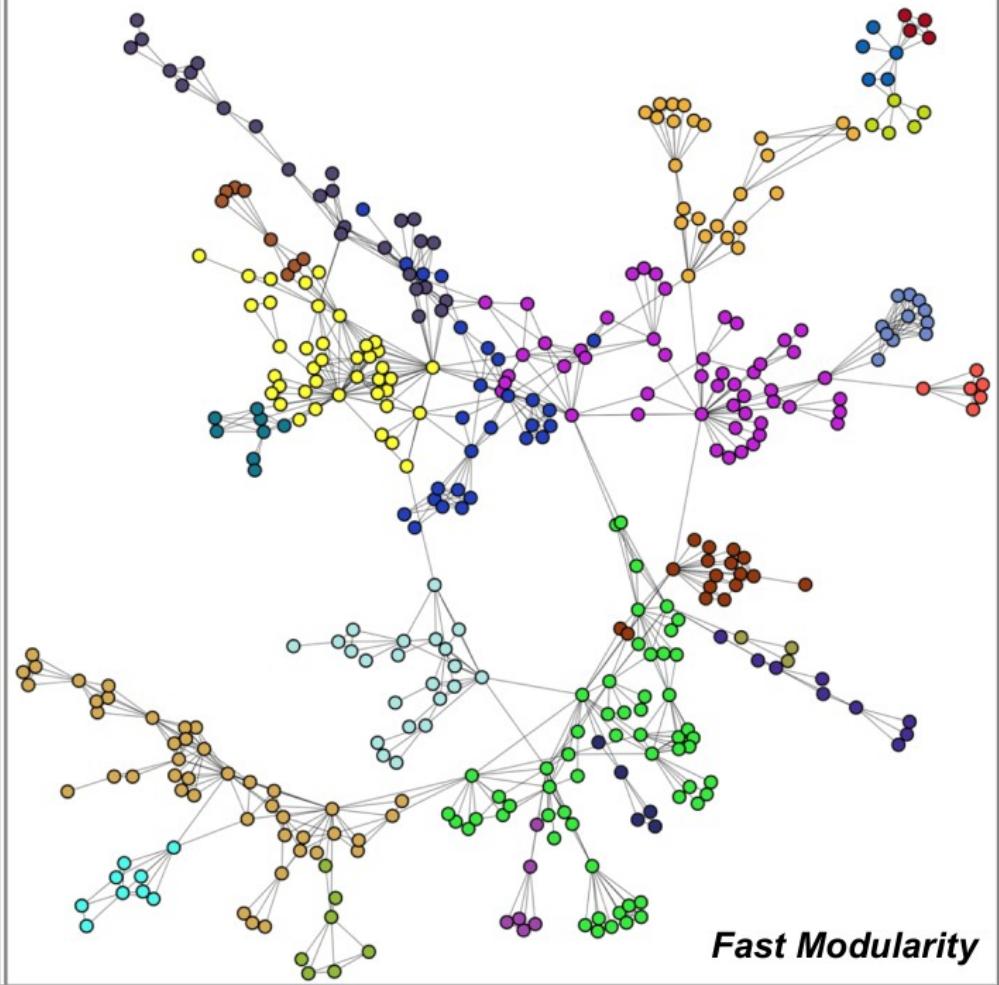
Community Structure

Roles and Communities

Roles



Communities



Henderson, et al., KDD 2012

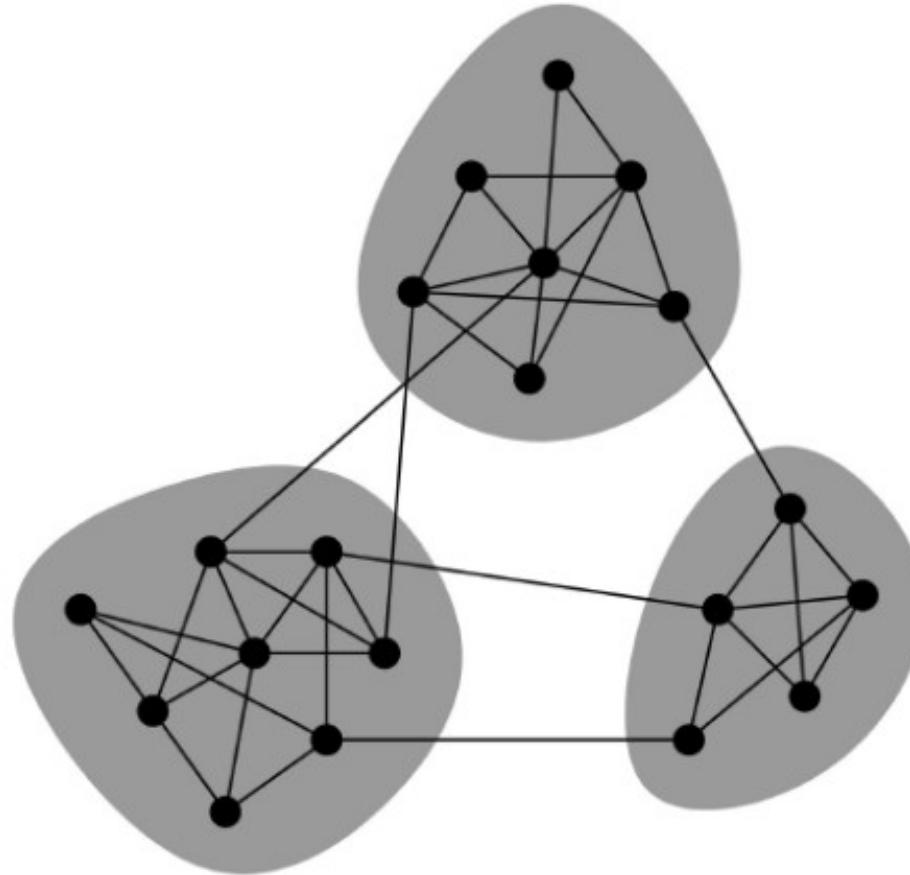
Nodes with different structural roles
(connector node, bridge node, etc.)

Clauset, et al., Phys. Rev. E 2004

Nodes belonging to the same
cluster/community

Networks and Communities

- We often think of networks “looking” like this:



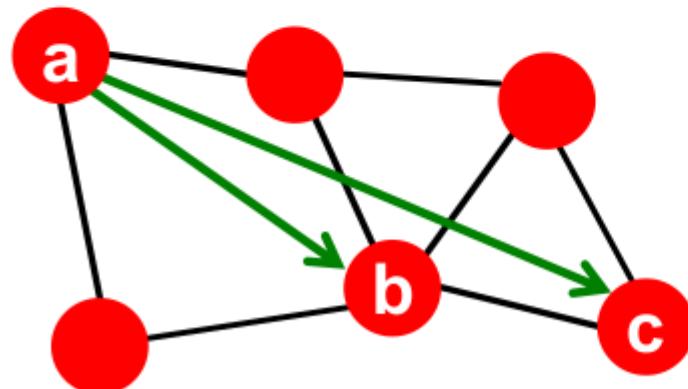
- What led to such a conceptual picture?

Networks: Flow of Information

- **How does information flow through the network?**
 - What structurally distinct roles do nodes play?
 - What roles do different links (“short” vs. “long”) play?
- **How do people find out about new jobs?**
 - Mark Granovetter, part of his PhD in 1960s
 - People find the information through personal contacts
- **But:** Contacts were often **acquaintances** rather than close friends
 - **This is surprising:** One would expect your friends to help you out more than casual acquaintances
- **Why is it that acquaintances are most helpful?**

Granovetter's Answer

- Two perspectives on friendships:
 - Structural: Friendships span different parts of the network
 - Interpersonal: Friendship between two people is either strong or weak
- Structural role: Triadic Closure

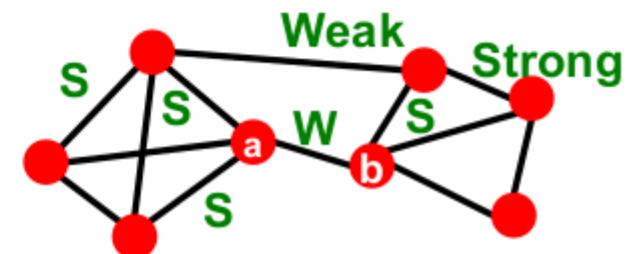


Which edge is more likely, a-b or a-c?

If two people in a network have a friend in common, then there is an increased likelihood they will become friends themselves.

Granovetter's Explanation

- Granovetter makes a connection between social and structural role of an edge
- First point: Structure
 - Structurally embedded edges are also socially strong
 - Long-range edges spanning different parts of the network are socially weak
- Second point: Information
 - Long-range edges allow you to gather information from different parts of the network and get a job
 - Structurally embedded edges are heavily redundant in terms of information access

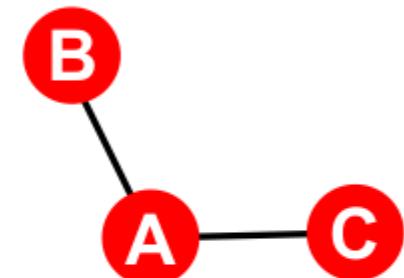


Triadic Closure

- **Triadic closure == High clustering coefficient**

Reasons for triadic closure:

- If **B** and **C** have a friend **A** in common, then:
 - **B** is more likely to meet **C**
 - (since they both spend time with **A**)
 - **B** and **C** trust each other
 - (since they have a friend in common)
 - **A** has incentive to bring **B** and **C** together
 - (since it is hard for **A** to maintain two disjoint relationships)
- **Empirical study by Bearman and Moody:**
 - Teenage girls with low clustering coefficient are more likely to contemplate suicide



Tie Strength in Real Data

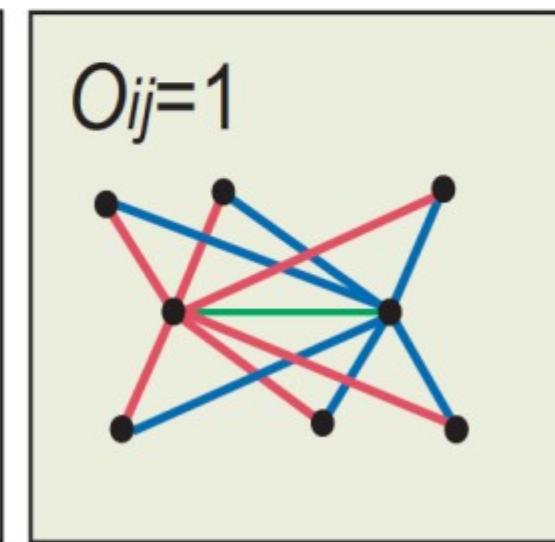
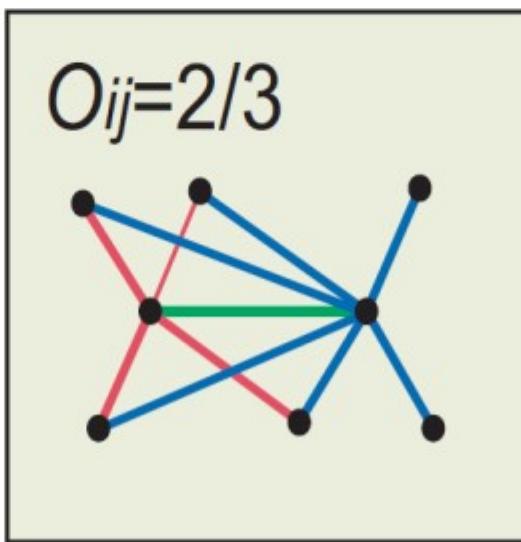
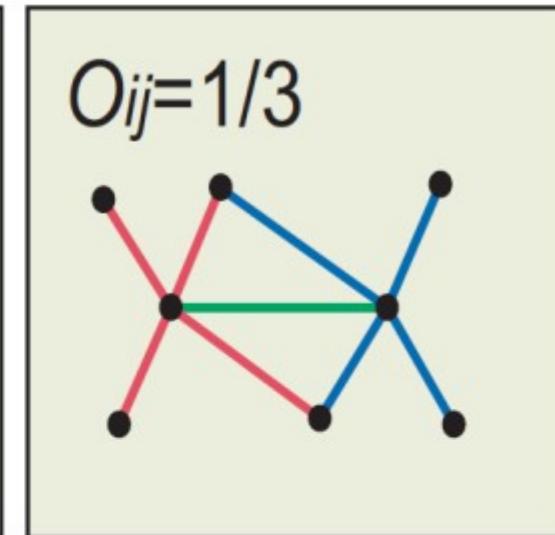
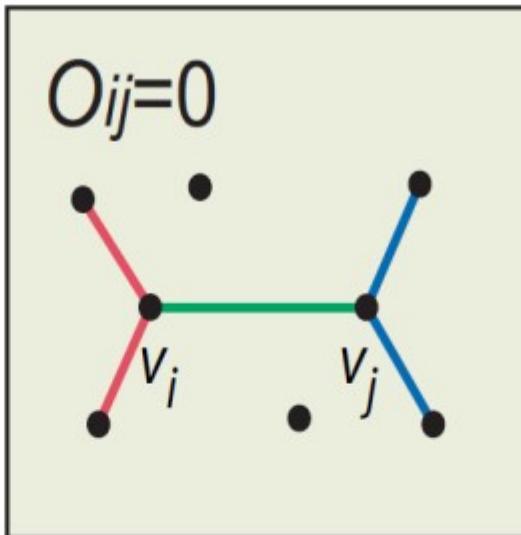
- For many years Granovetter's theory was not tested
- But, today we have large who-talks-to-whom graphs:
 - Email, Messenger, Cell phones, Facebook
- Onnela et al. 2007:
 - Cell-phone network of 20% of country's population
 - Edge strength: # phone calls

Neighborhood Overlap

- **Edge overlap:**

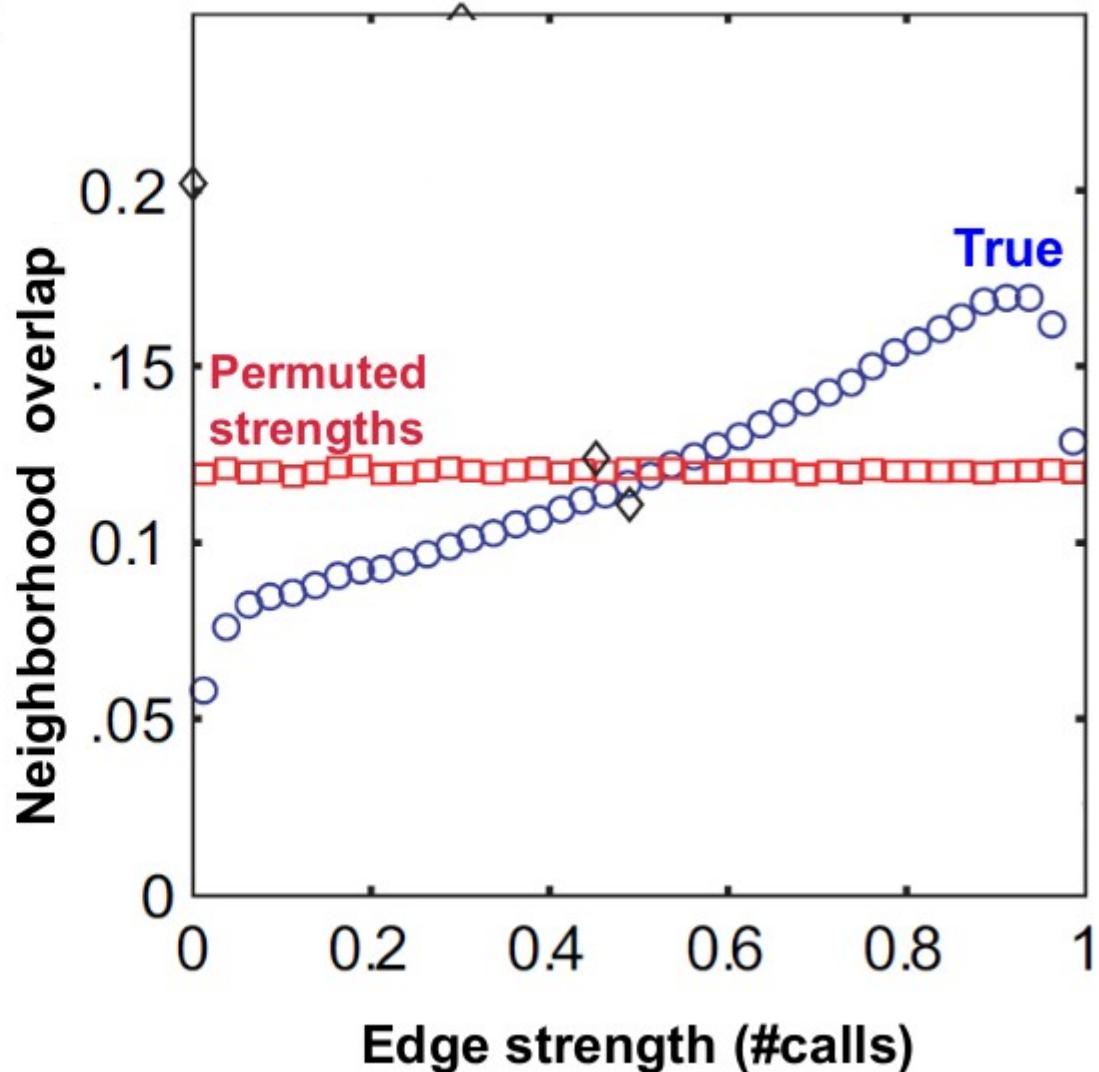
$$O_{ij} = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$$

- $N(i)$... a set of neighbors of node i
- **Overlap = 0** when an edge is a **local bridge**

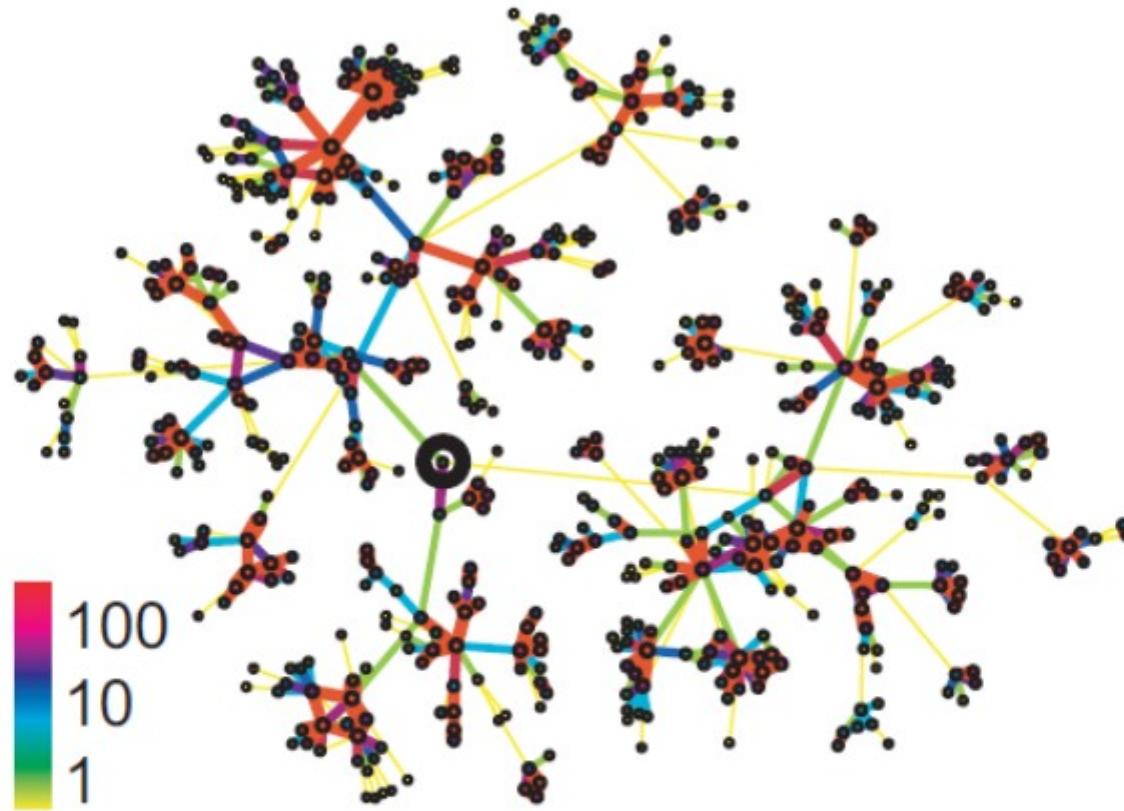


Phones: Edge Overlap vs Strength

- Cell-phone network
- Observation:
 - Highly used links have high overlap!
- Legend:
 - True: The data
 - Permutated strengths: Keep the network structure but randomly reassign edge strengths

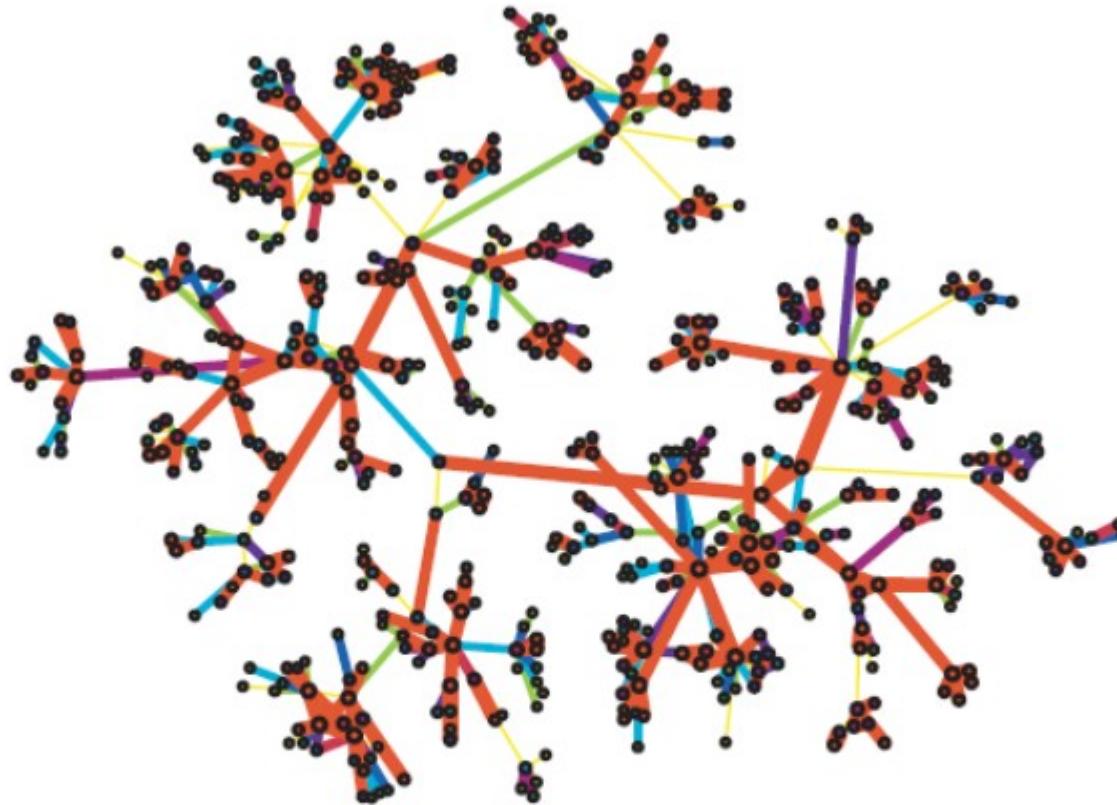


Real Net, Real Tie Strengths



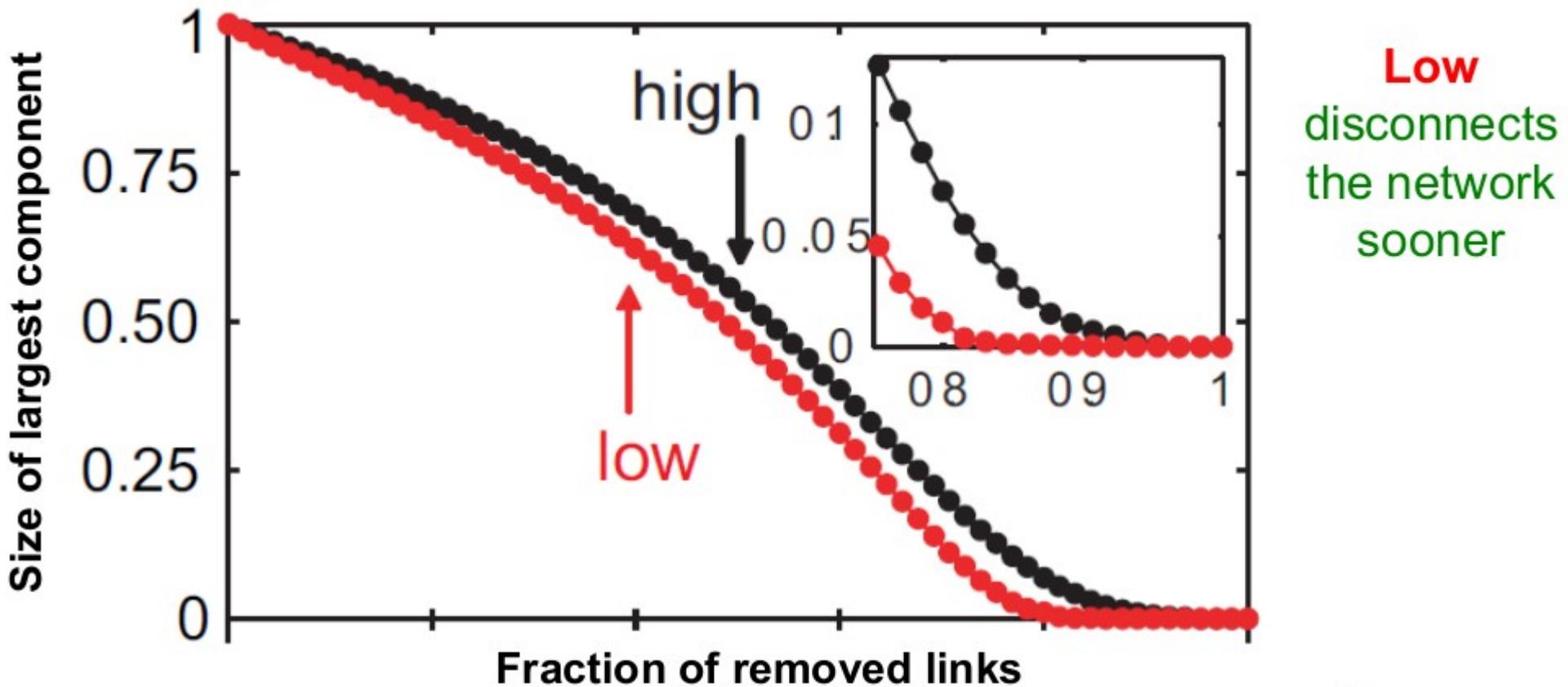
- **Real edge strengths in mobile call graph**
 - Strong ties are more embedded (have higher overlap)

Real Net, Permuted Tie Strengths

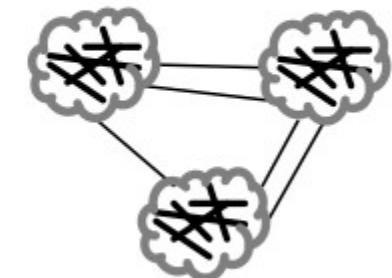


- Same network, same set of edge strengths
but now **strengths are randomly shuffled**

Link Removal by Strength

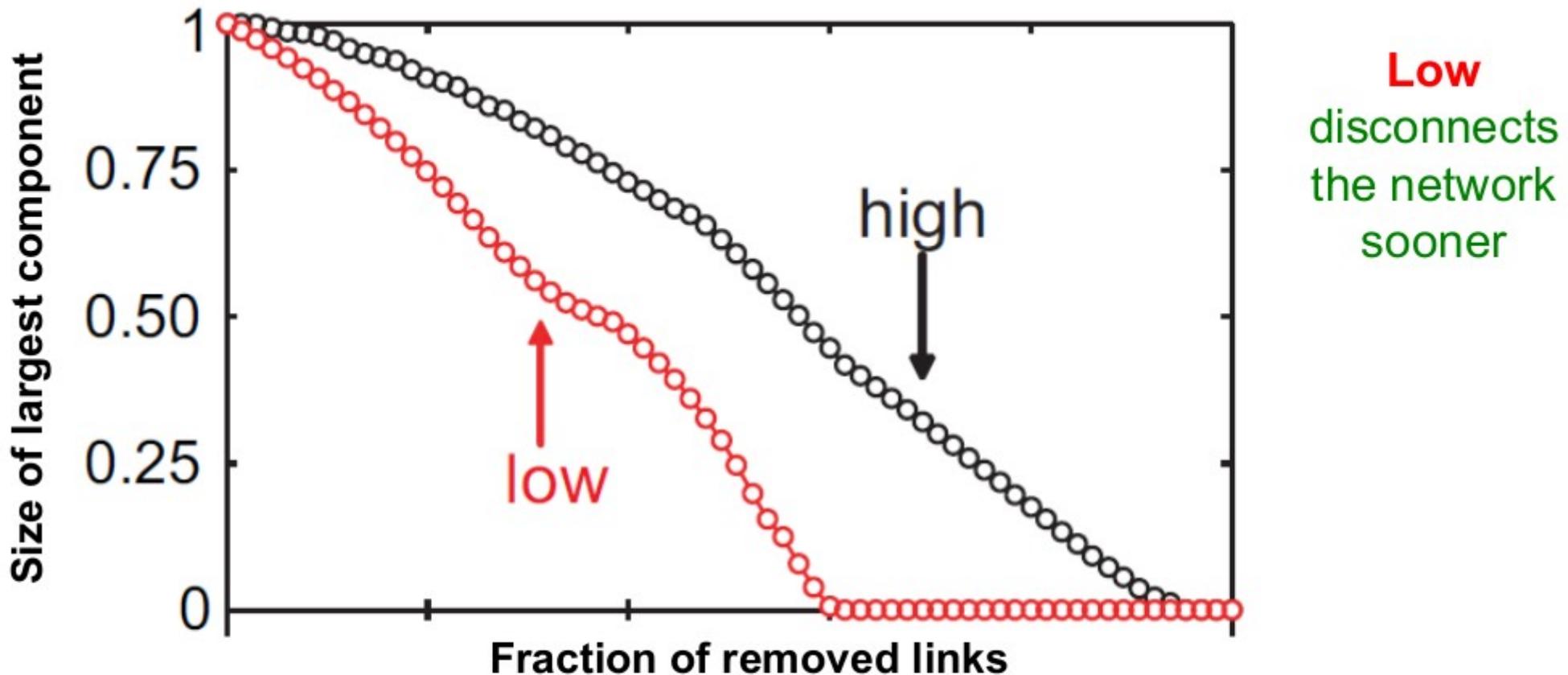


- Removing links by **strength (#calls)**
 - Low to high
 - High to low

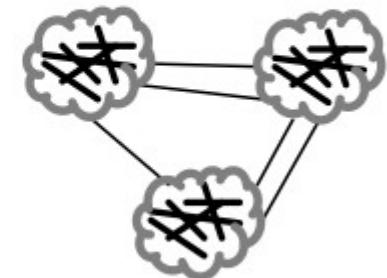


Conceptual picture
of network structure

Link Removal by Overlap



- Removing links based on **overlap**
 - Low to high
 - High to low

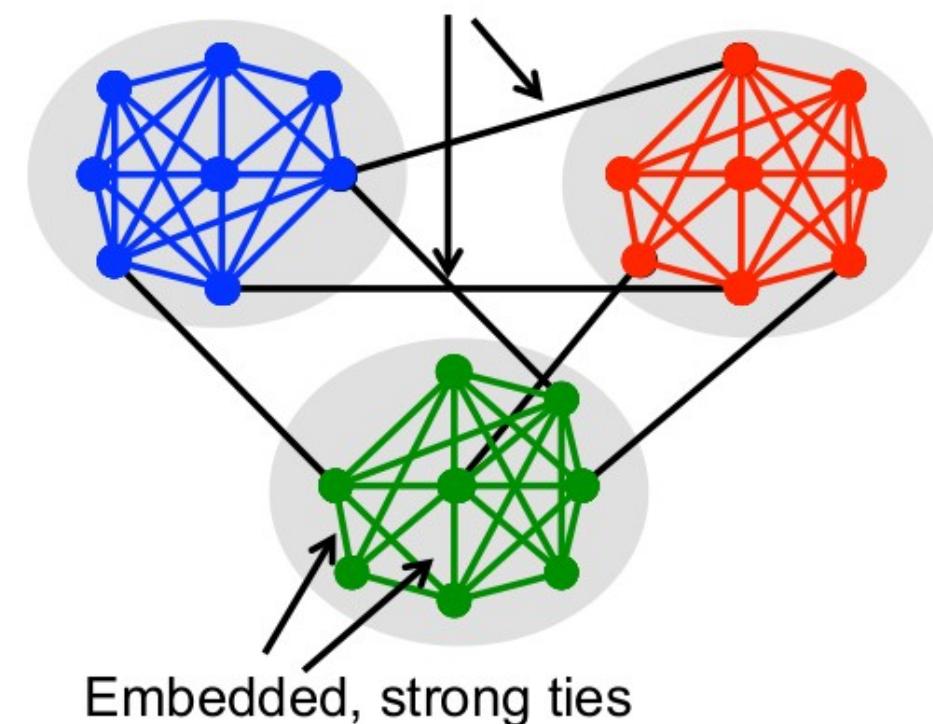


Conceptual picture
of network structure

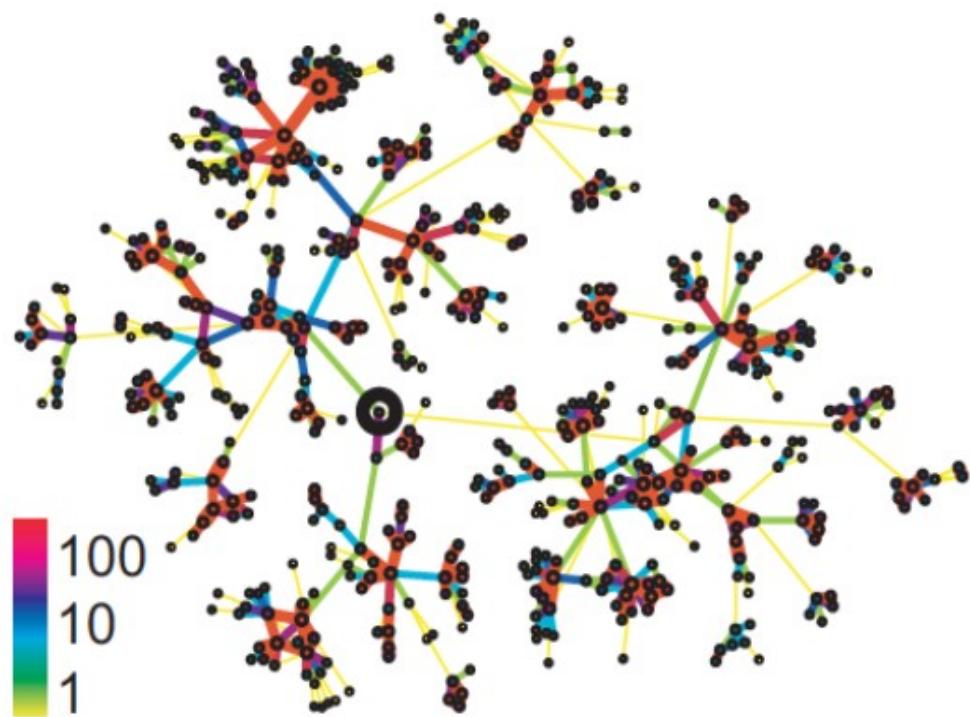
Closing the Loop

- We often think of (social) networks as having the following structure

Long-range, weak ties



Embedded, strong ties

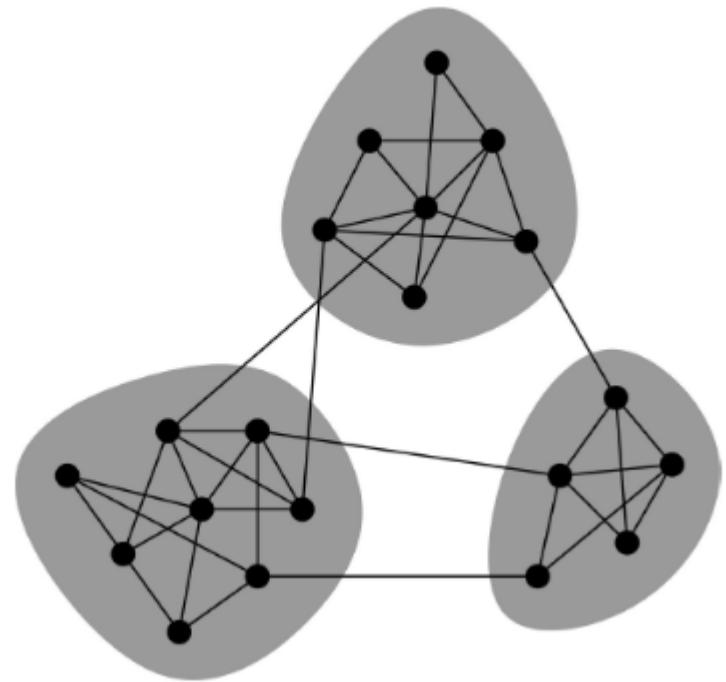


- Conceptual picture supported by Granovetter's **strength of weak ties**

Network Communities

Network Communities

- Granovetter's theory suggest that networks are composed of **tightly connected sets of nodes**



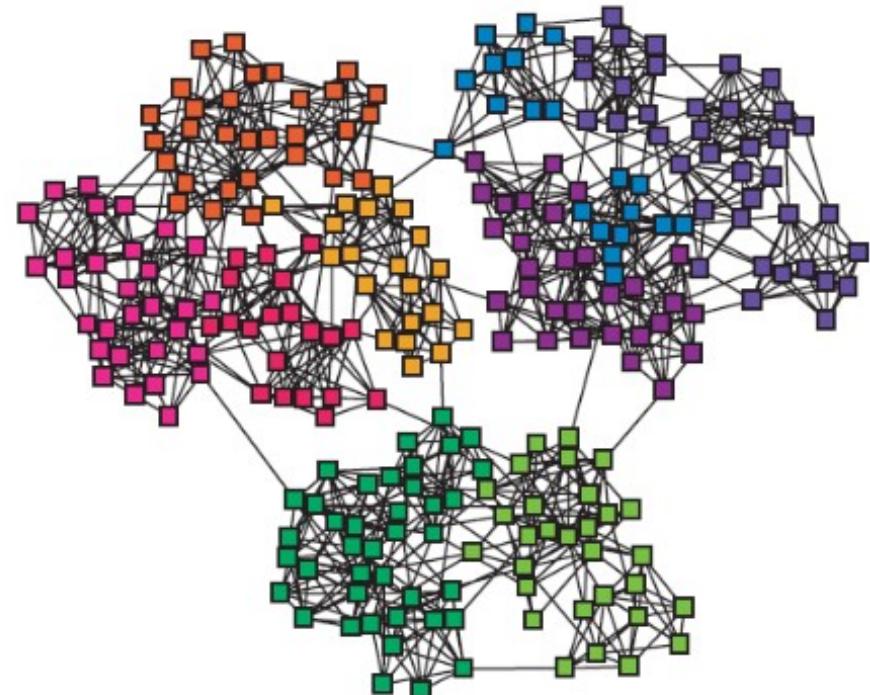
Communities, clusters, groups, modules

- **Network communities:**

- Sets of nodes with **lots of internal** connections and **few external** ones (to the rest of the network).

Finding Network Communities

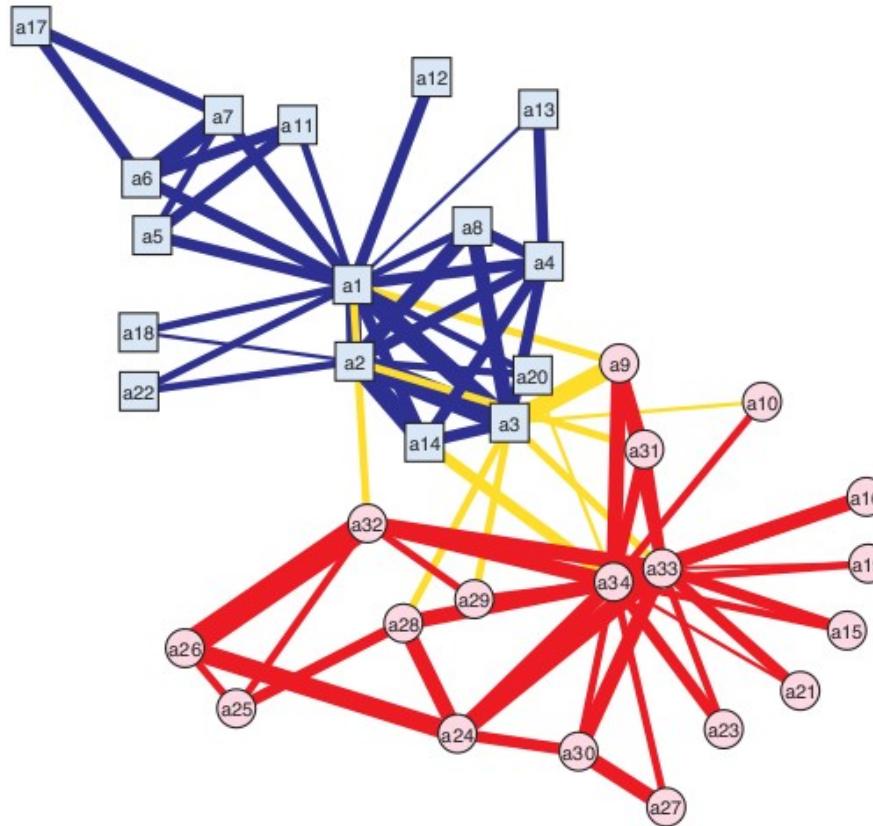
- **How to automatically find such densely connected groups of nodes?**
- Ideally such automatically detected clusters would then correspond to real groups
- **For example:**



Communities, clusters,
groups, modules

Zachary's karate club

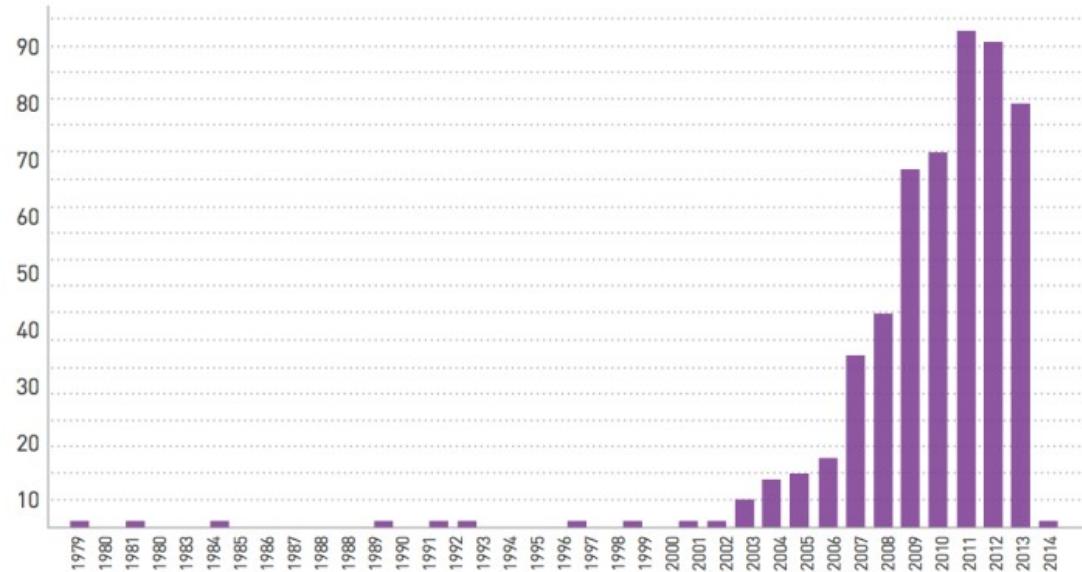
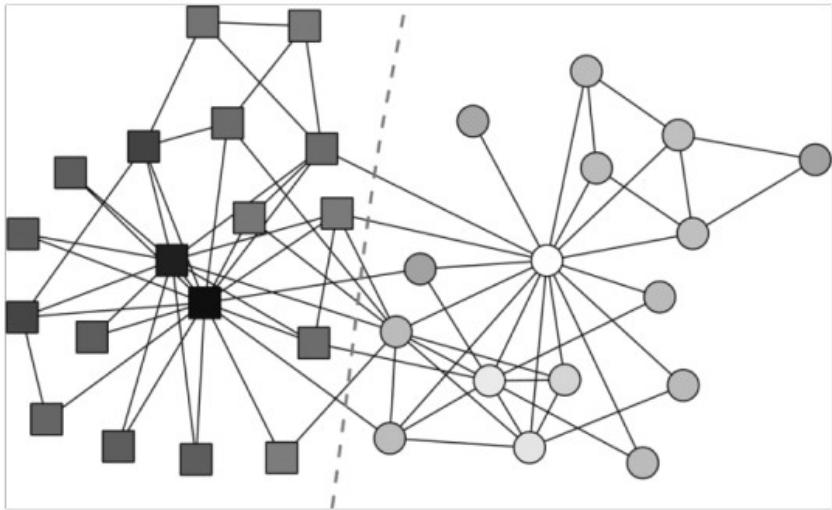
- ▶ Social interactions among members of a karate club in the 70s



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

Zachary's karate club

Citation history
of the Zachary's Karate club paper



Zachary's karate club Club!

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

Chris Moore (9 May 2013).

Mason Porter (NetSci, June 2013).

Yong-Yeol Ahn (Oxford University, July 2013)

Marián Boguñá (ECCS, September 2013).

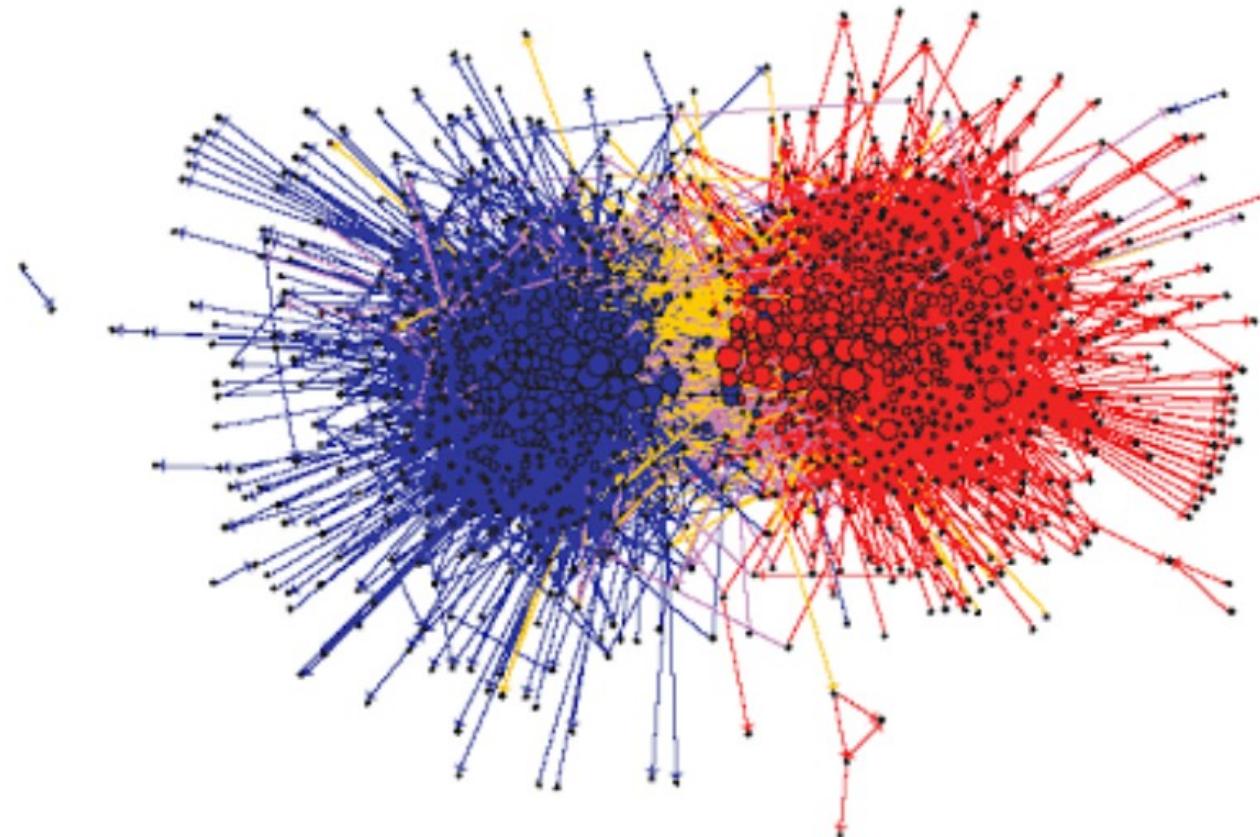
Mark Newman (Netsci, June 2014)



<http://networkkarate.tumblr.com/>

Political blogs

- ▶ The political blogosphere for the US 2004 presidential election



- ▶ Community structure of **liberal** and **conservative** blogs is apparent
 - ⇒ People have a stronger tendency to interact with “equals”

Electrical power grid

- ▶ Split power network into areas with minimum inter-area **interactions**

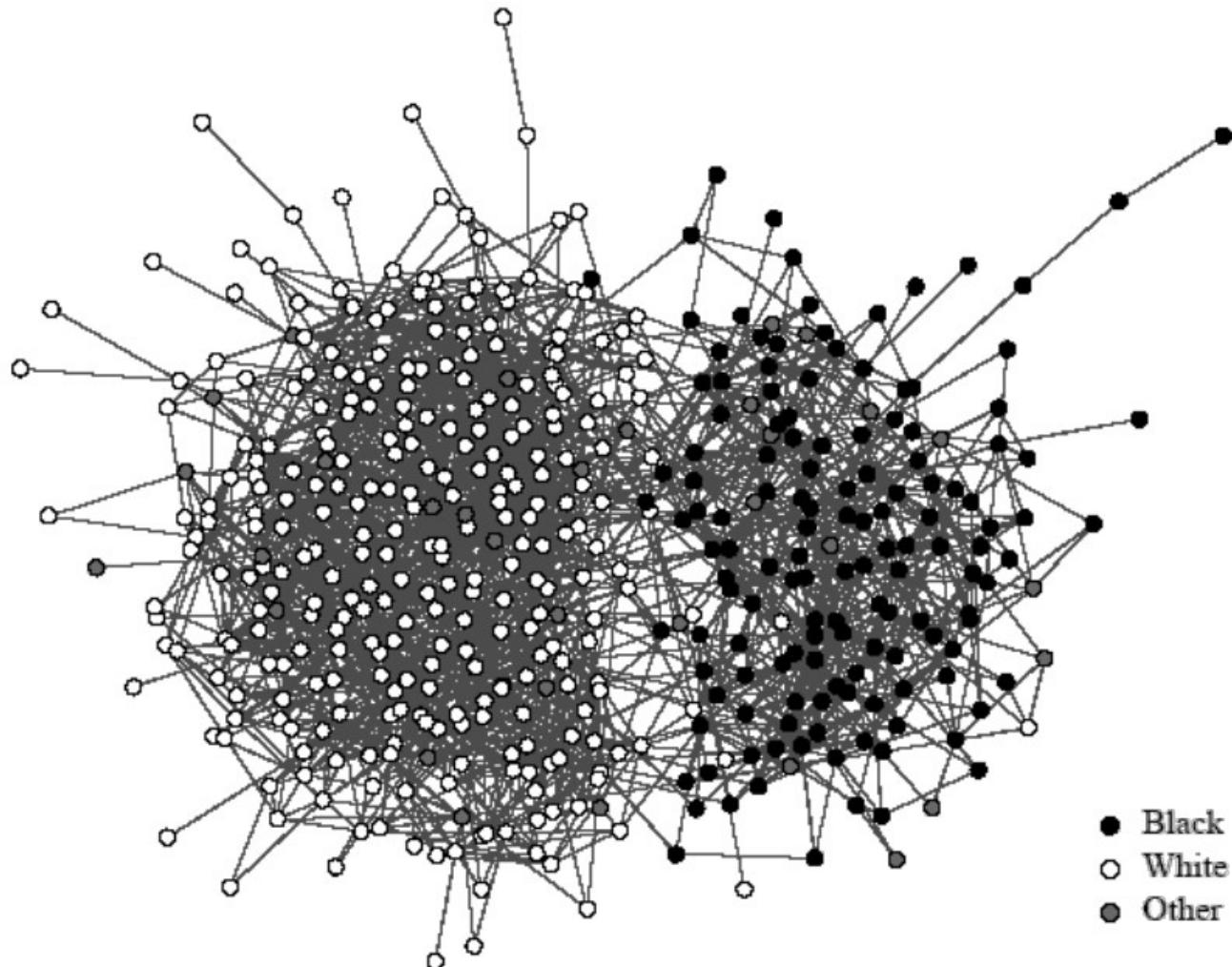


- ▶ **Applications:**

- ▶ Decide control areas for distributed power system state estimation
- ▶ Parallel computation of power flow
- ▶ Controlled islanding to prevent spreading of blackouts

High-school students

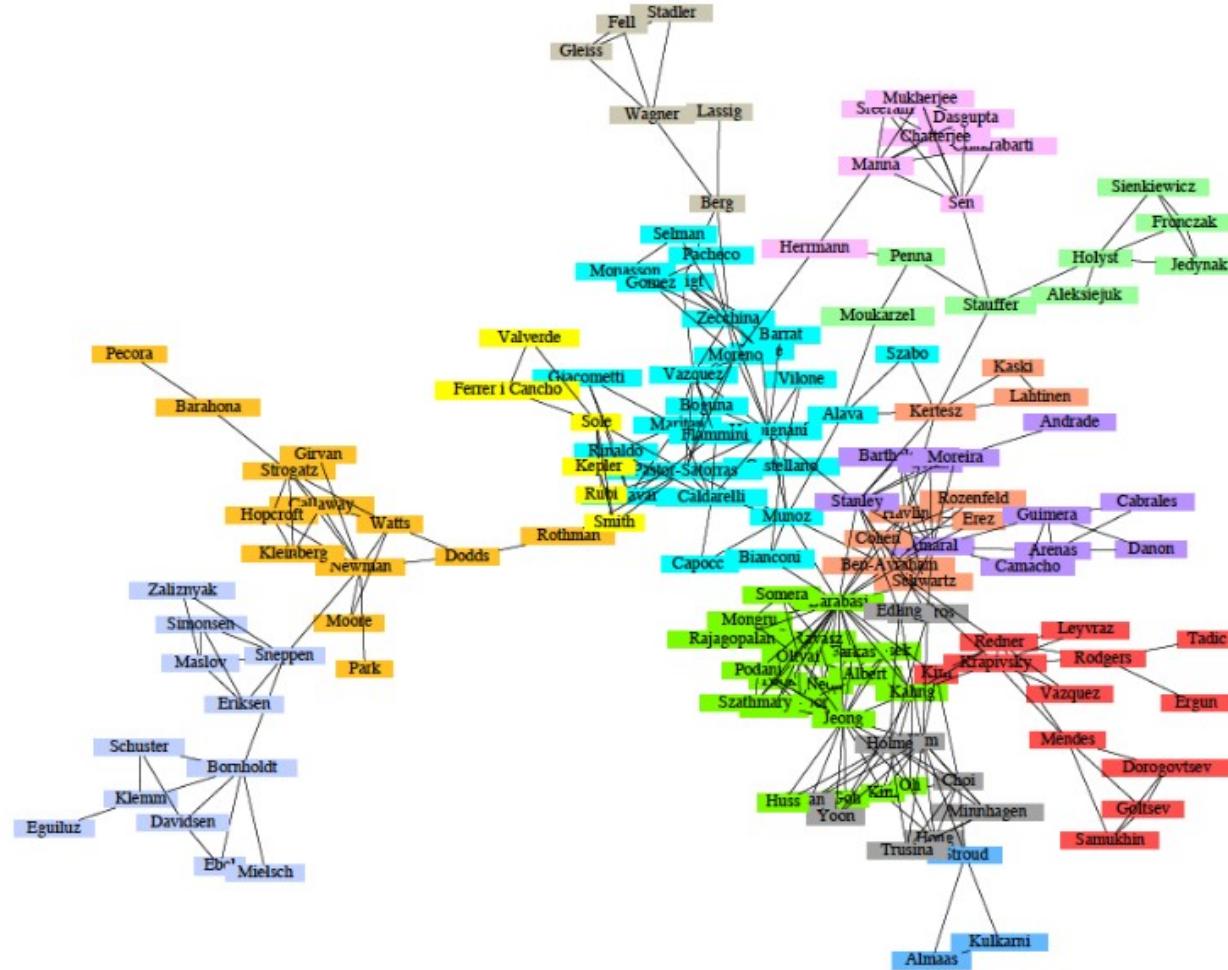
- ▶ Network of social interactions among high-school students



- ▶ Strong **assortative mixing**, with race as latent characteristic

Physicists working on NetSci

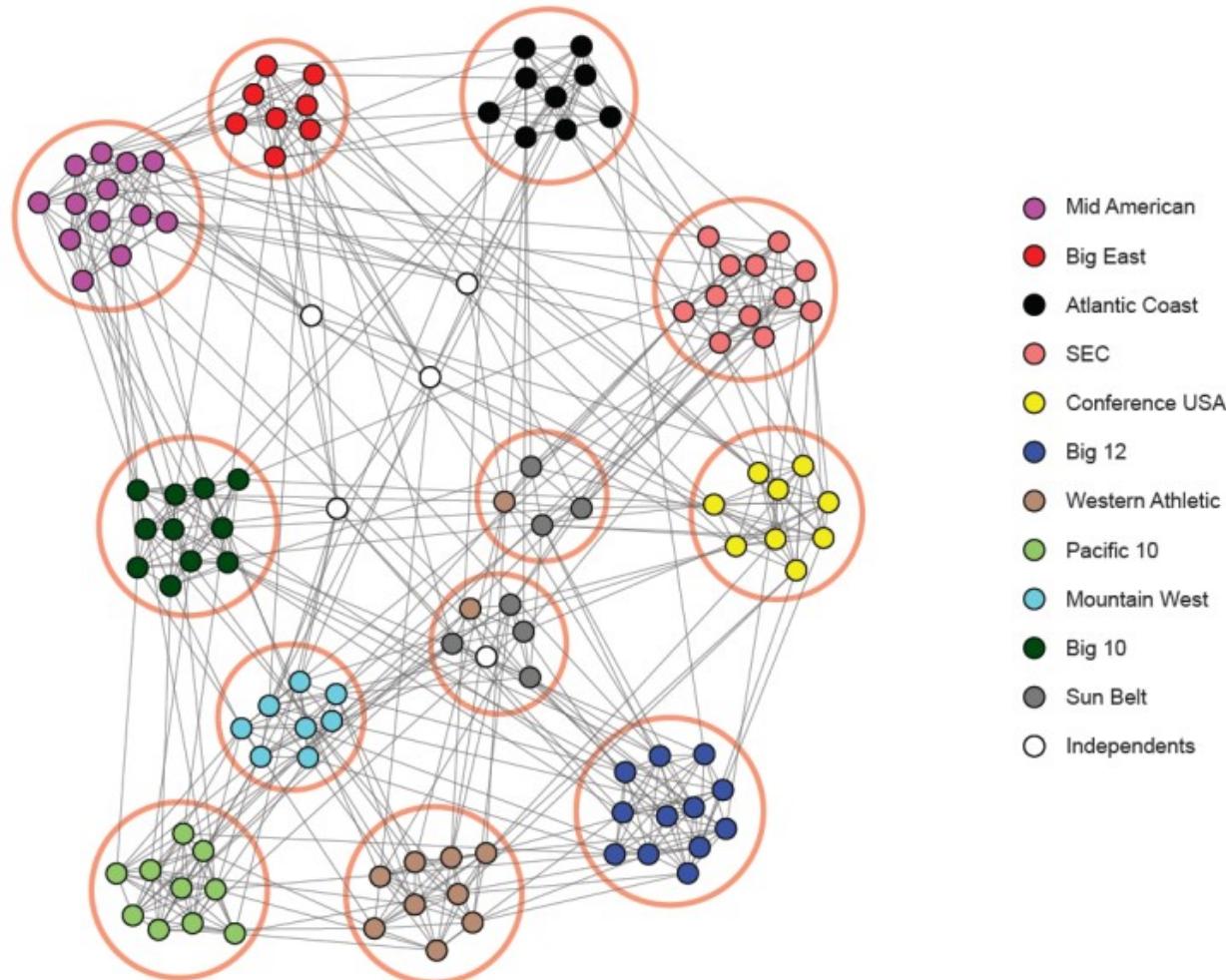
- ▶ Coauthorship network of physicists publishing networks' research



- ▶ Tightly-knit subgroups are evident from the network structure

College football

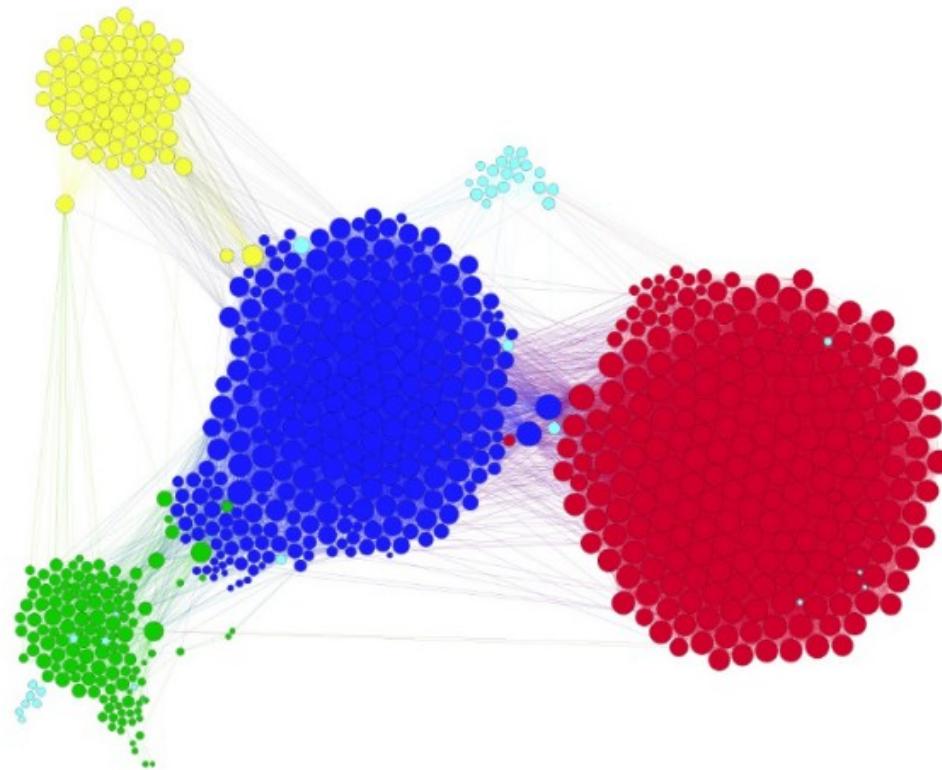
- Vertices are NCAA football teams, edges are games during Fall'00



- Communities are the NCAA conferences and independent teams

Facebook friendships

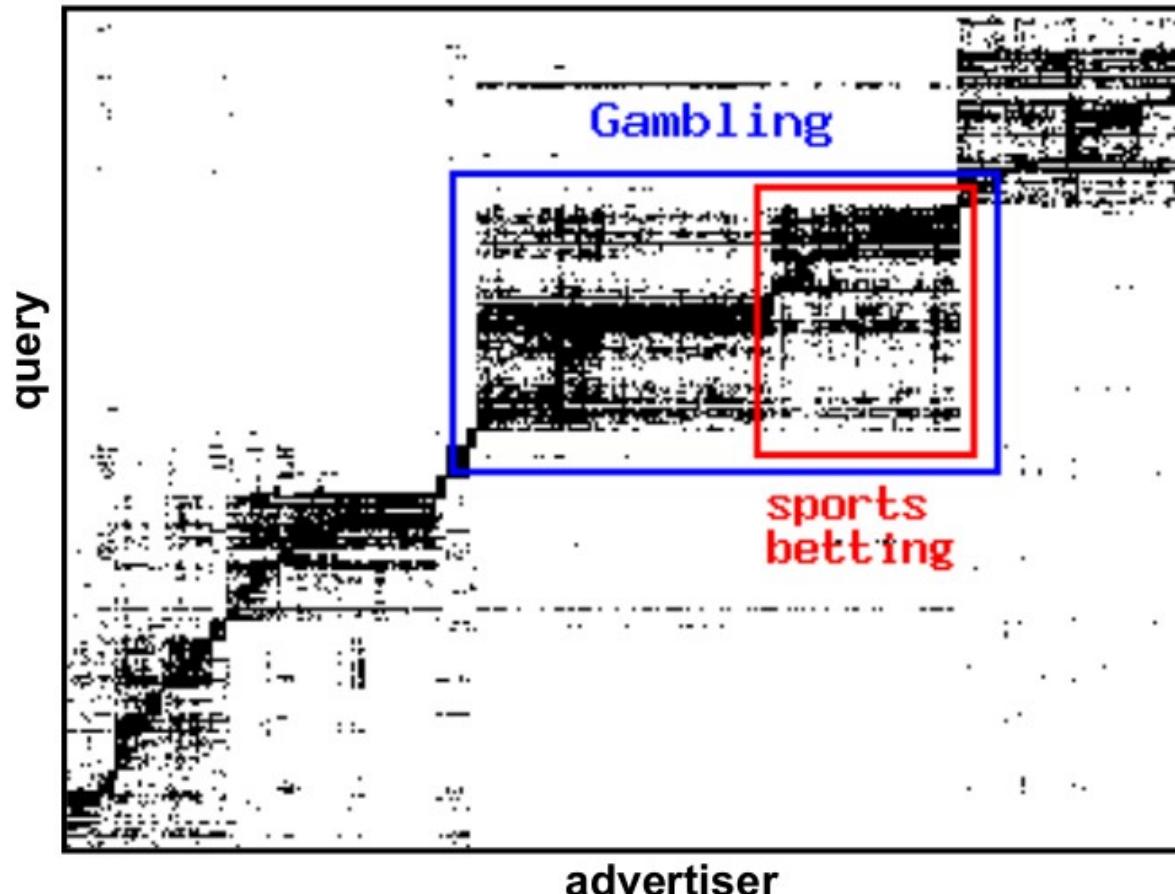
- ▶ Facebook egonet with 744 vertices and 30K edges



- ▶ Asked “ego” to identify social circles to which friends belong
 - ⇒ Company, high-school, basketball club, squash club, family

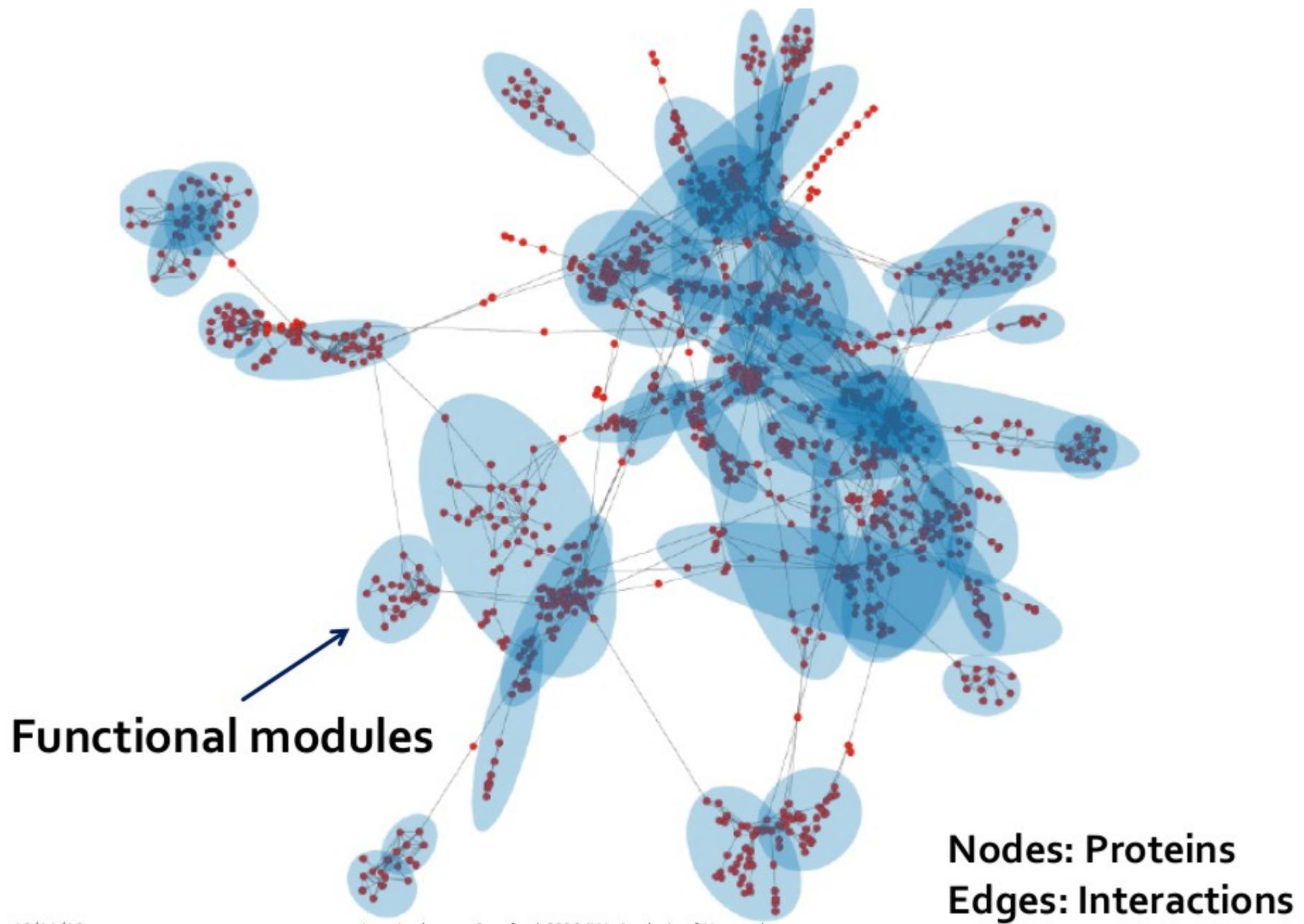
Micro-Markets in Sponsored Search

Find micro-markets by partitioning the “query-to-advertiser” graph in web search:

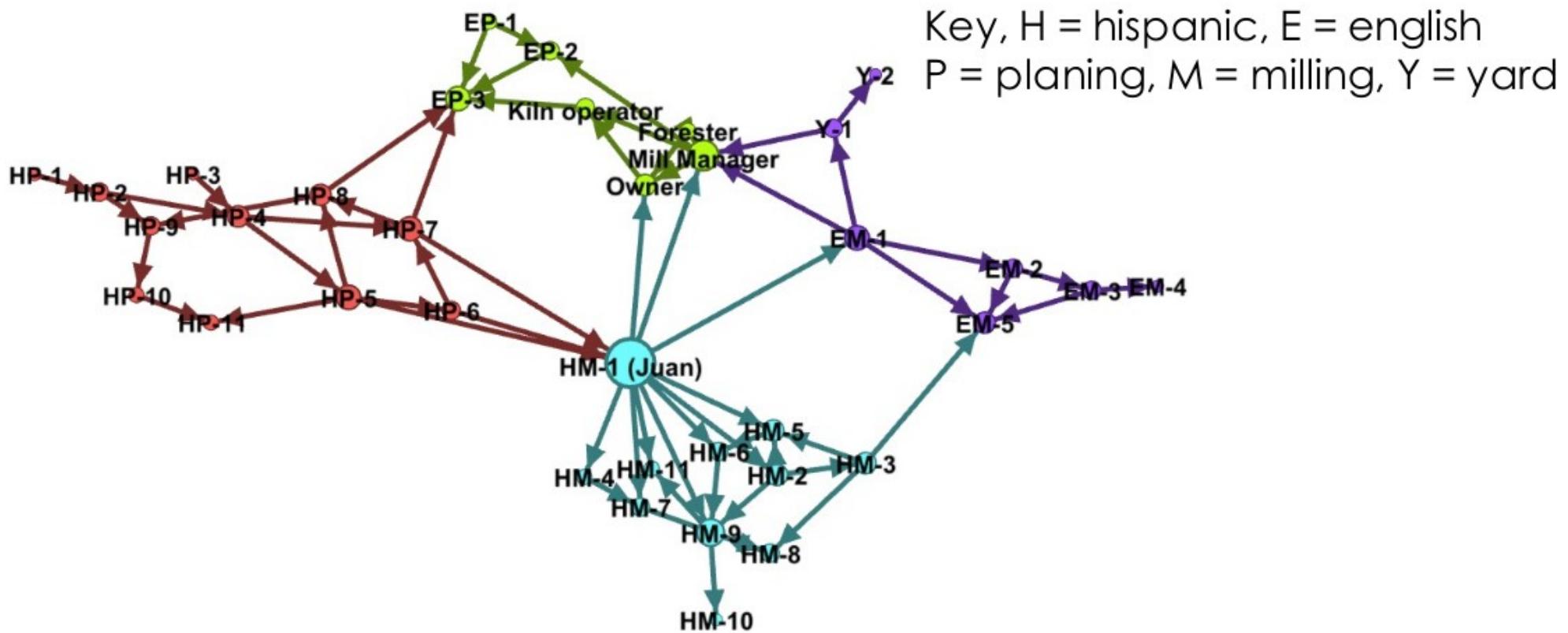


Nodes: advertisers and queries/keywords; Edges: Advertiser advertising on a keyword.

Protein-Protein Interaction



Why look for community structure?

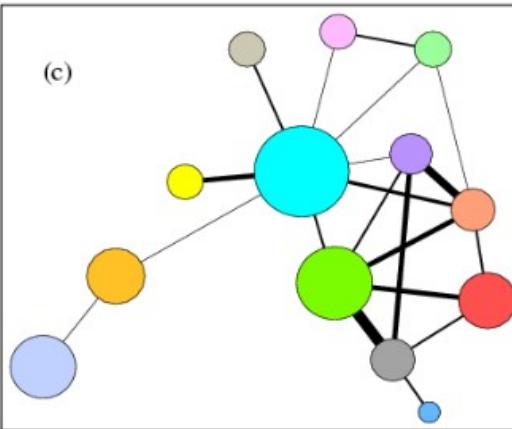
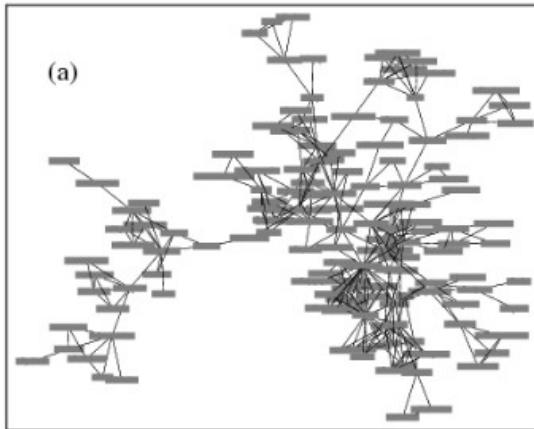


- The management at the sawmill was having difficulty persuading the workers to adopt a new plan, even though everyone would benefit. In particular the Hispanic workers (H) were reluctant to agree. The management called in a sociologist who mapped out who talked to whom regularly. Then they suggested that the management talk to Juan and have him talk to the Hispanic workers. It was a success, promptly everyone was on board with the new plan. Why?

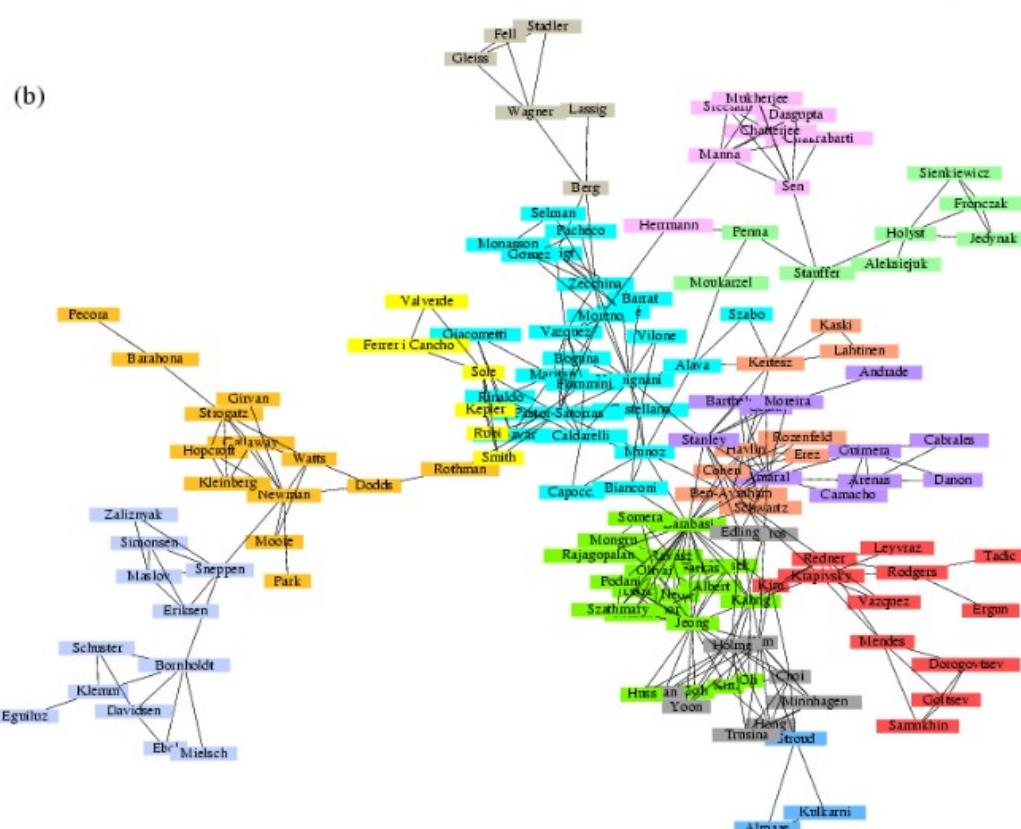
Why: gain understanding

- ❑ Gain understanding of networks
 - ❑ Discover communities of practice
 - ❑ Measure isolation of groups
 - ❑ Understand opinion dynamics / adoption

Why: Visualize



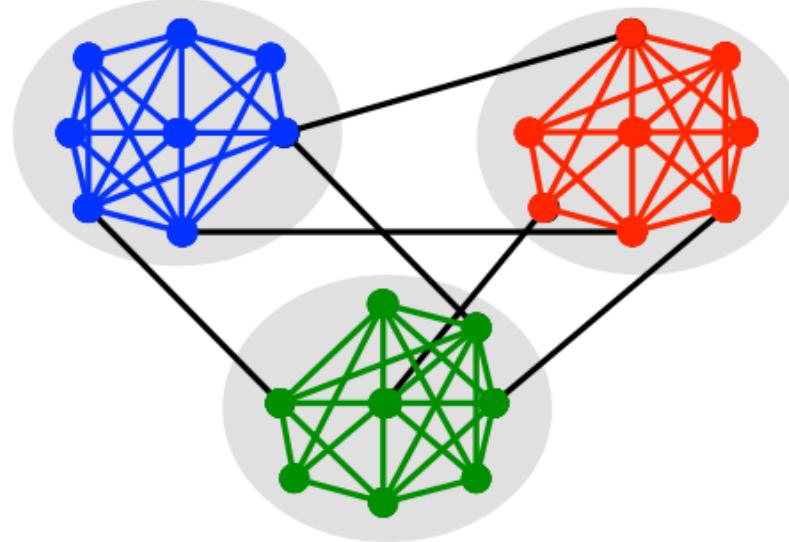
Why do it:
visualize



■ Communities
help to
“aggregate”
network
data

Unveiling network communities

- ▶ Nodes in real-world networks organize into **communities**
Ex: families, clubs, political organizations, proteins by function, ...



- ▶ Community (a.k.a. group, cluster, module) members are:
 - ⇒ Well connected among themselves
 - ⇒ Relatively well separated from the rest
- ▶ Exhibit high **cohesiveness** w.r.t. the underlying relational patterns
- ▶ **Q:** How can we **automatically identify** such cohesive subgroups?

Community detection and graph partitioning

- ▶ Community detection is a challenging clustering problem
 - C1) No consensus on the structural definition of community
 - C2) Node subset selection often intractable
 - C3) Lack of ground-truth for validation
- ▶ Useful for exploratory analysis of network data
 - Ex: clues about social interactions, content-related web pages

Graph partitioning

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

- ▶ Criterion: number of edges between groups is minimized (more soon)
 - Ex: task-processor assignment for load balancing
- ▶ Number and sizes of groups unspecified in community detection
 - ⇒ Identify the natural fault lines along which a network separates

Graph partitioning is hard

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

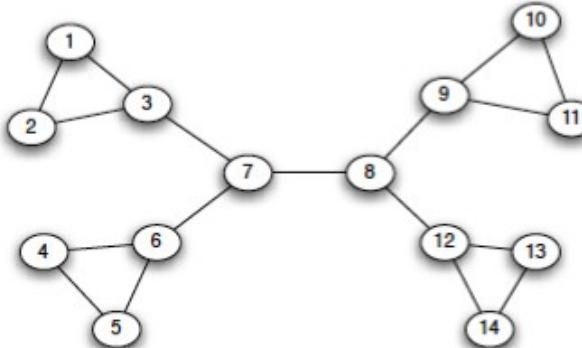
Number of ways to partition V : $\binom{N_v}{N_v/2} \approx \frac{2^{N_v}}{\sqrt{N_v}}$

- ⇒ Used Stirling's formula $N_v! \approx \sqrt{2\pi N_v} (N_v/e)^{N_v}$
- ⇒ Exhaustive search intractable beyond toy small-sized networks

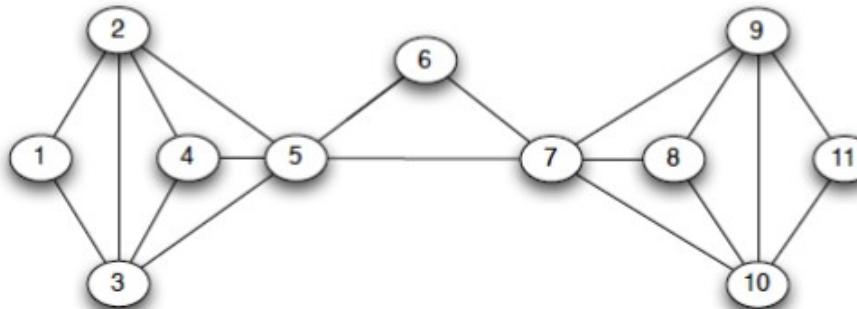
- No smart (i.e., polynomial time) algorithm, **NP-hard problem**
 - ⇒ Seek good heuristics, e.g., relaxations of natural criteria

Strength of weak ties motivation

- ▶ Local bridges connect weakly interacting parts of the network



- ▶ **Q:** What about removing those to reveal communities?

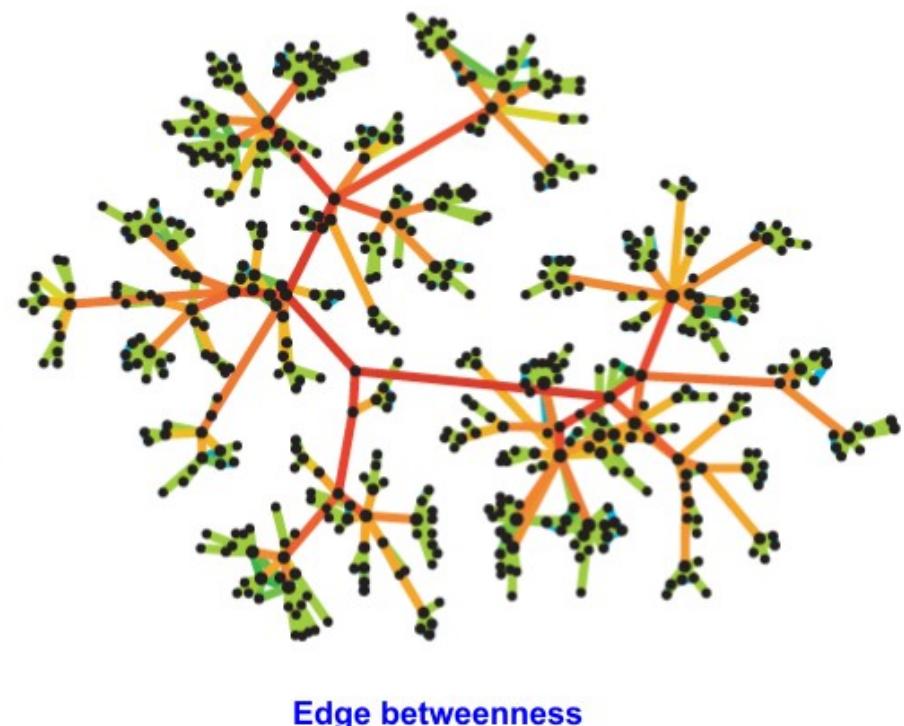
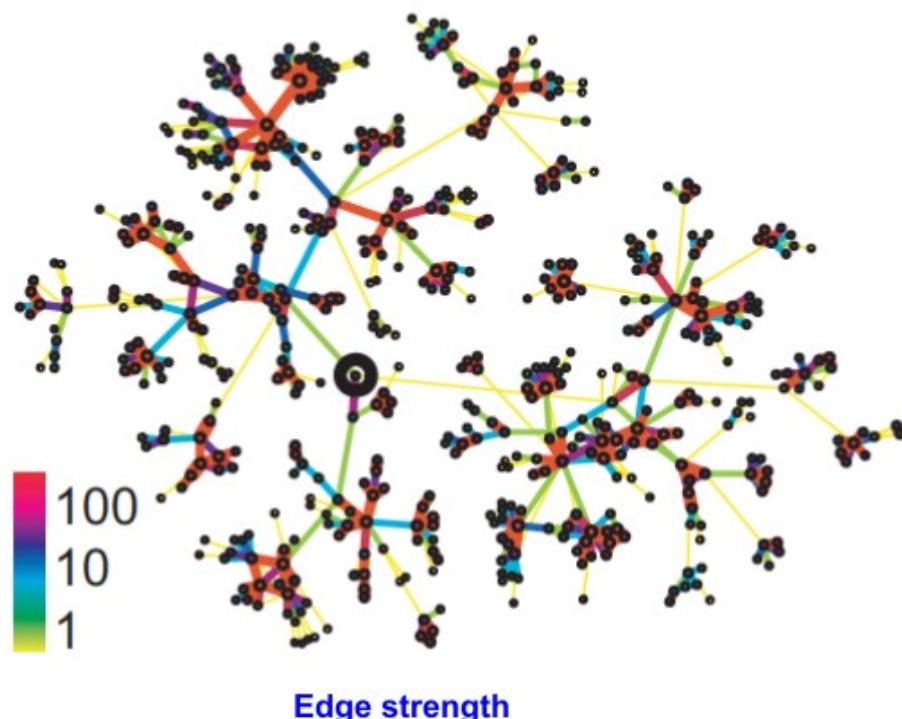


- ▶ **Challenges**

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

Edge betweenness centrality

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

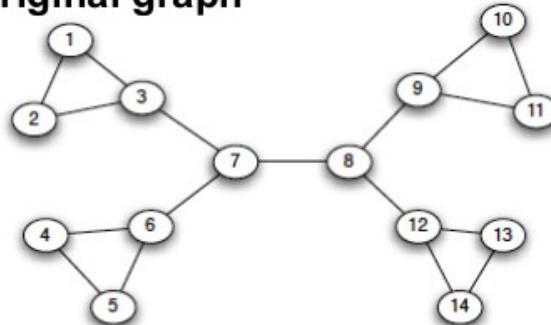


Girvan-Newman's method

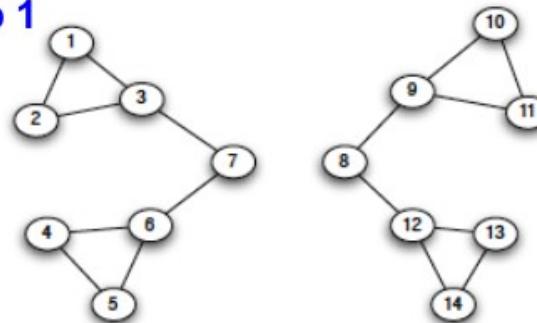
- ▶ Girvan-Newmann's method extremely simple conceptually
 - ⇒ Find and remove “spanning links” between cohesive subgroups
- ▶ **Algorithm:** Repeat until there are no edges left
 - ⇒ Calculate the betweenness centrality $c_{Be}(e)$ of all edges
 - ⇒ Remove edge(s) with highest $c_{Be}(e)$
- ▶ Connected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

Example: The algorithm in action

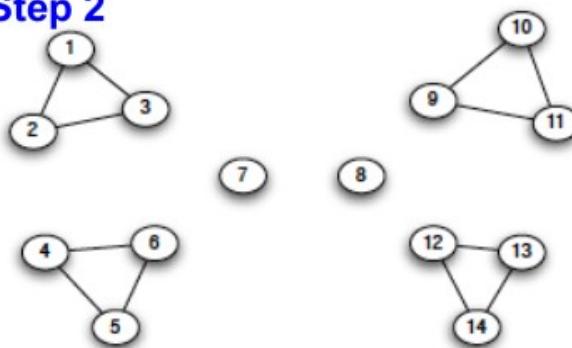
Original graph



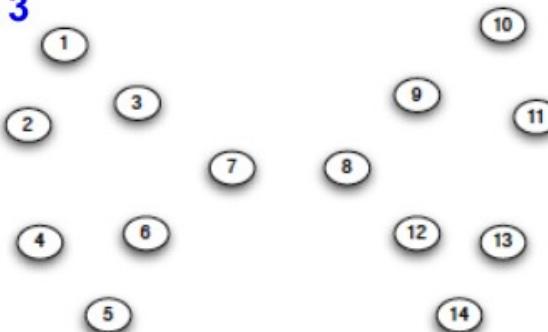
Step 1



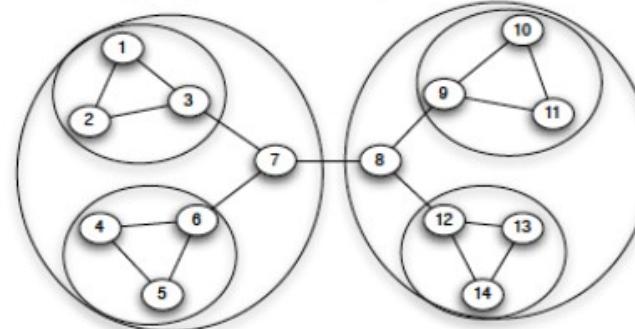
Step 2



Step 3

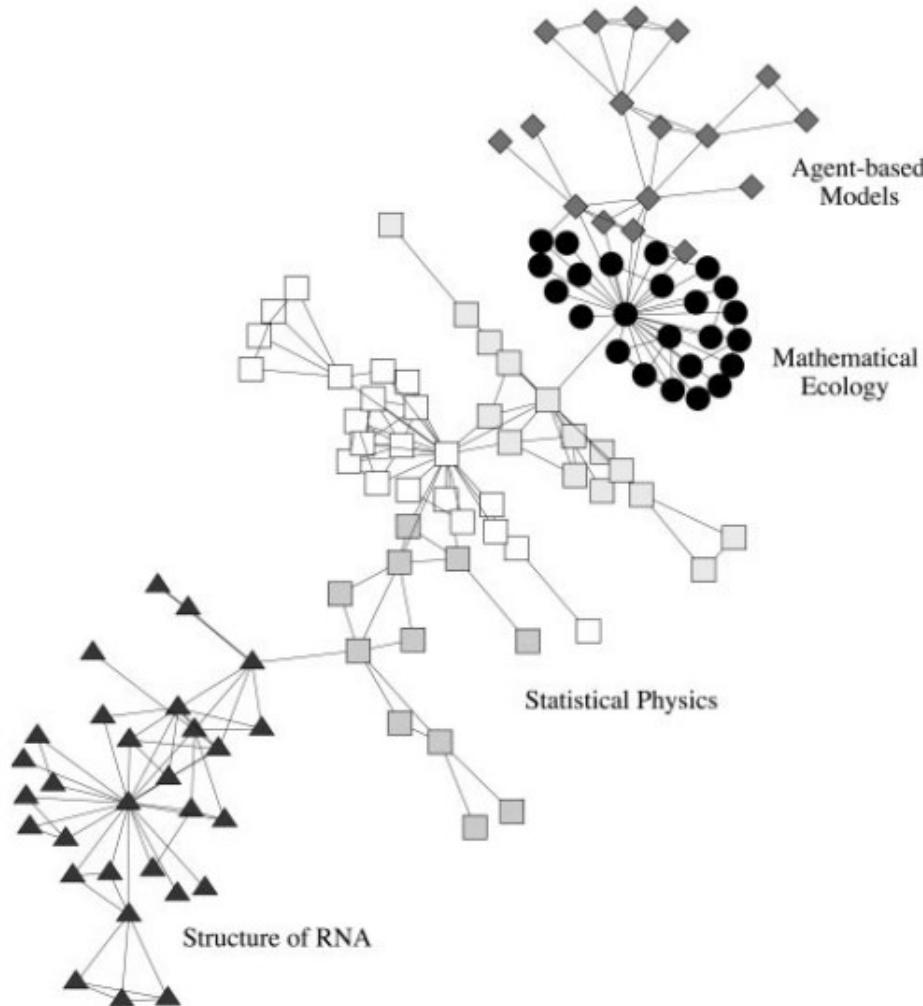


Nested graph decomposition



Scientific collaboration network

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



- Communities found can be traced to different disciplines

Hierarchical clustering

- ▶ Greedy approach to iteratively modify successive candidate partitions
 - ▶ **Agglomerative**: successive coarsening of partitions through merging
 - ▶ **Divisive**: successive refinement of partitions through splitting
- ▶ Per step, partitions are modified in a way that minimizes a cost
 - ▶ Measures of (dis)similarity x_{ij} between pairs of vertices v_i and v_j
 - ▶ **Ex**: Euclidean distance dissimilarity
- ▶ Method returns an entire hierarchy of nested partitions of the graph
⇒ Can range fully from $\{\{v_1\}, \dots, \{v_{N_v}\}\}$ to V

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

Agglomerative clustering

- An **agglomerative hierarchical clustering algorithm** proceeds as follows
 - S1:** Choose a dissimilarity metric and compute it for all vertex pairs
 - S2:** Assign each vertex to a group of its own
 - S3:** Merge the pair of groups with smallest dissimilarity
 - S4:** Compute the dissimilarity between the new group and all others
 - S5:** Repeat from S3 until all vertices belong to a single group
- Need to define **group dissimilarity** from pairwise vertex counterparts
 - **Single linkage:** group dissimilarity x_{G_i, G_j}^{SL} follows single most dissimilar pair

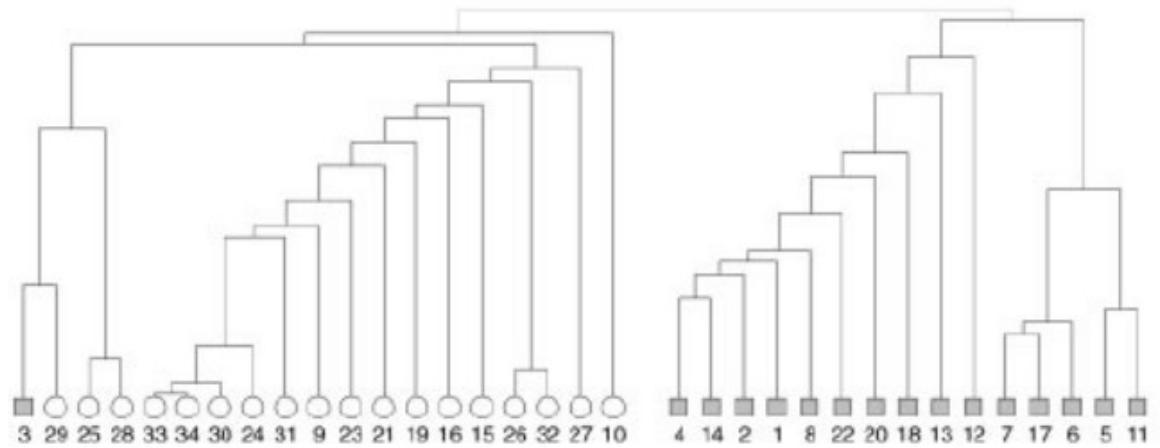
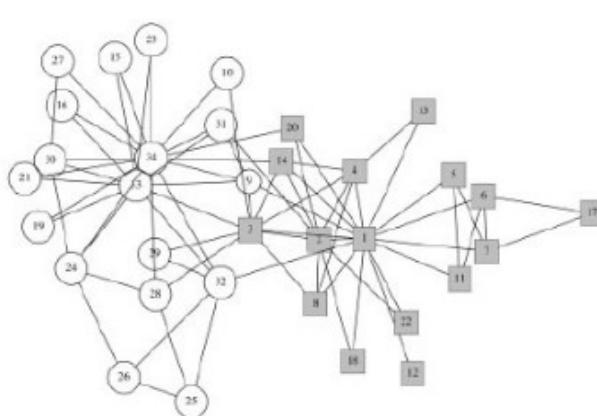
$$x_{G_i, G_j}^{SL} = \max_{u \in G_i, v \in G_j} x_{uv}$$

- **Complete linkage:** every vertex pair highly dissimilar to have high x_{G_i, G_j}^{CL}

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

Dendrogram

- ▶ Hierarchical partitions often represented with a **dendrogram**
- ▶ Shows groups found in the network at all algorithmic steps
⇒ Split the network at different resolutions
- ▶ **Ex:** Girvan-Newman's algorithm for the Zachary's karate club



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

Modularity

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

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

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

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

\Rightarrow Null model: randomize edges, preserving degree distribution

Expected connectivity among nodes

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

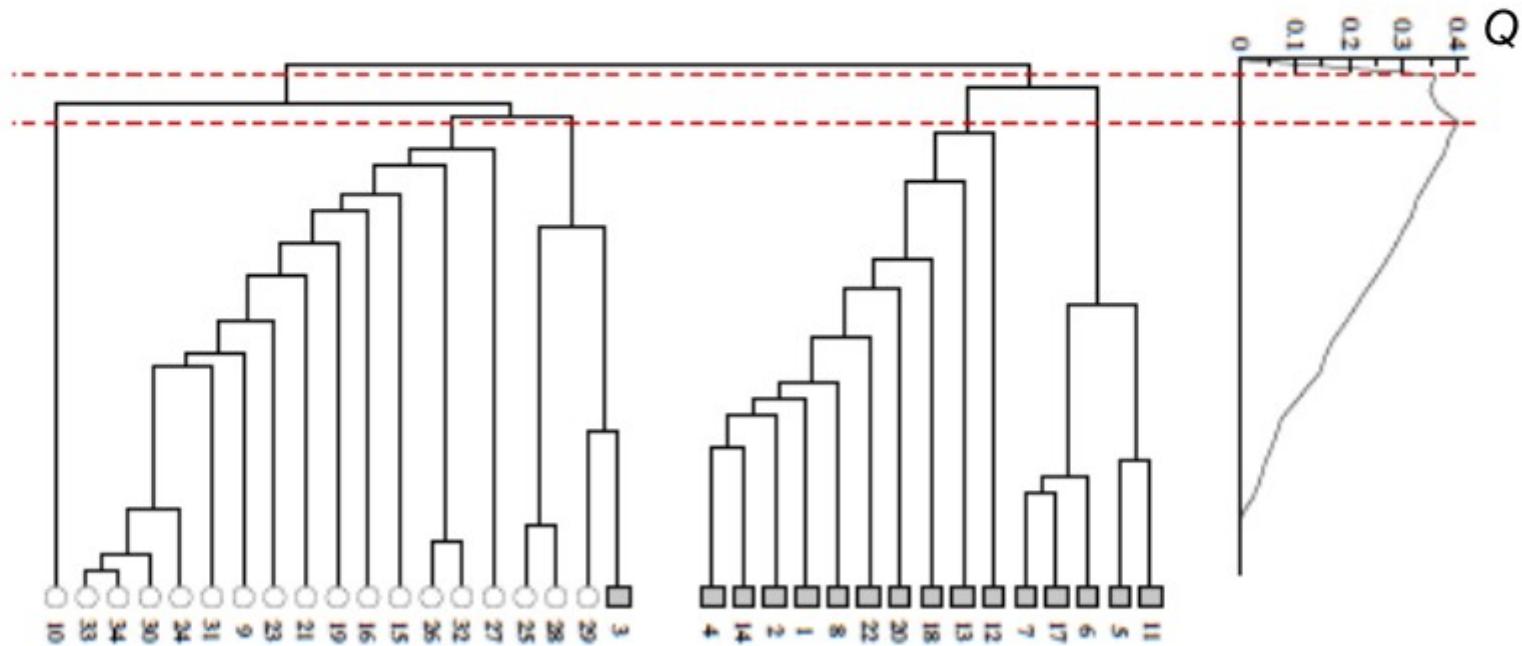


- ▶ Probability spoke i_k connected to j is $\frac{d_j}{2N_e - 1} \approx \frac{d_j}{2N_e}$, hence

$$\begin{aligned} P((i,j) \in E) &= P\left(\bigcup_{i_k=1}^{d_i} \{\text{spoke } i_k \text{ connected to } j\}\right) \\ &= \sum_{i_k=1}^{d_i} P(\text{spoke } i_k \text{ connected to } j) = \frac{d_i d_j}{2N_e} \end{aligned}$$

Assessing clustering quality

- ▶ Can evaluate the modularity of each partition in a dendrogram
⇒ Maximum value gives the “best” community structure
- ▶ Ex: Girvan-Newman’s algorithm for the Zachary’s karate club



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

Modularity: another look

- Modularity of partitioning S of graph G :
 - $Q \propto \sum_{s \in S} [(\# \text{ edges within group } s) - (\text{expected } \# \text{ edges within group } s)]$
 - $$Q(G, S) = \underbrace{\frac{1}{2m}}_{\text{Normalizing const.: } -1 < Q < 1} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$
 $A_{ij} = 1 \text{ if } i \rightarrow j,$
 0 else
- Modularity values take range $[-1, 1]$
 - It is positive if the number of edges within groups exceeds the expected number
 - Q greater than **0.3-0.7** means **significant community structure**

Modularity: another look

- Consider edges that fall within a community or between a community and the rest of the network
- Define modularity:

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w)$$

adjacency matrix probability of an edge between two vertices is proportional to their degrees
if vertices are in the same community

- For a random network, $Q = 0$
 - the number of edges within a community is no different from what you would expect

Modularity: another look

$$Q(G, S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

Equivalently modularity can be written as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

- A_{ij} represents the edge weight between nodes i and j ;
- k_i and k_j are the sum of the weights of the edges attached to nodes i and j , respectively;
- $2m$ is the sum of all of the edge weights in the graph;
- c_i and c_j are the communities of the nodes; and
- δ is an indicator function

Idea: We can identify communities by maximizing modularity

Louvain Algorithm

Louvain Algorithm

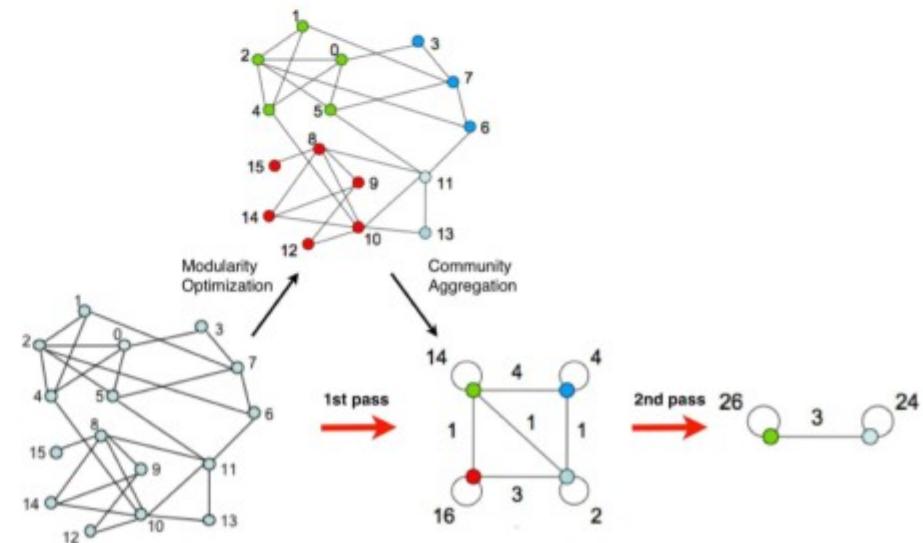
- **Greedy algorithm** for community detection
 - $O(n \log n)$ run time
- Supports weighted graphs
- Provides hierarchical partitions
- Widely utilized to **study large networks** because:
 - Fast
 - Rapid convergence properties
 - High modularity output (i.e., “better communities”)

“Fast unfolding of communities in large networks” Blondel et al. (2008)

Louvain Algorithm: at high level

- Louvain algorithm **greedily maximizes** modularity
- **Each pass is made of 2 phases:**
 - **Phase 1:** Modularity is **optimized** by allowing only local changes of communities
 - **Phase 2:** The identified communities are **aggregated** in order to build a new network of communities
 - **Goto Phase 1**

The passes are repeated **iteratively** until no increase of modularity is possible!



Louvain: 1st phase (partitioning)

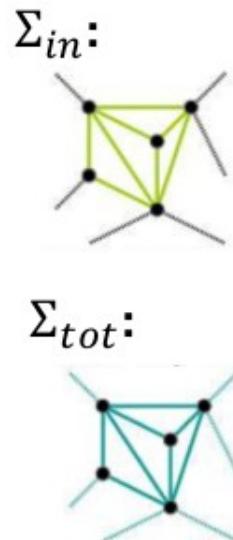
- Put each node in a graph into a **distinct community** (one node per community)
- For each node i , the algorithm performs two calculations:
 - Compute the modularity gain (ΔQ) when putting node i into the community of some neighbor j
 - Move i to a community of node j that yields the largest gain ΔQ
- The loop runs until no movement yields a gain

Louvain: Modularity Gain

What is ΔQ if we move node i to community C ?

$$\Delta Q(i \rightarrow C) = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

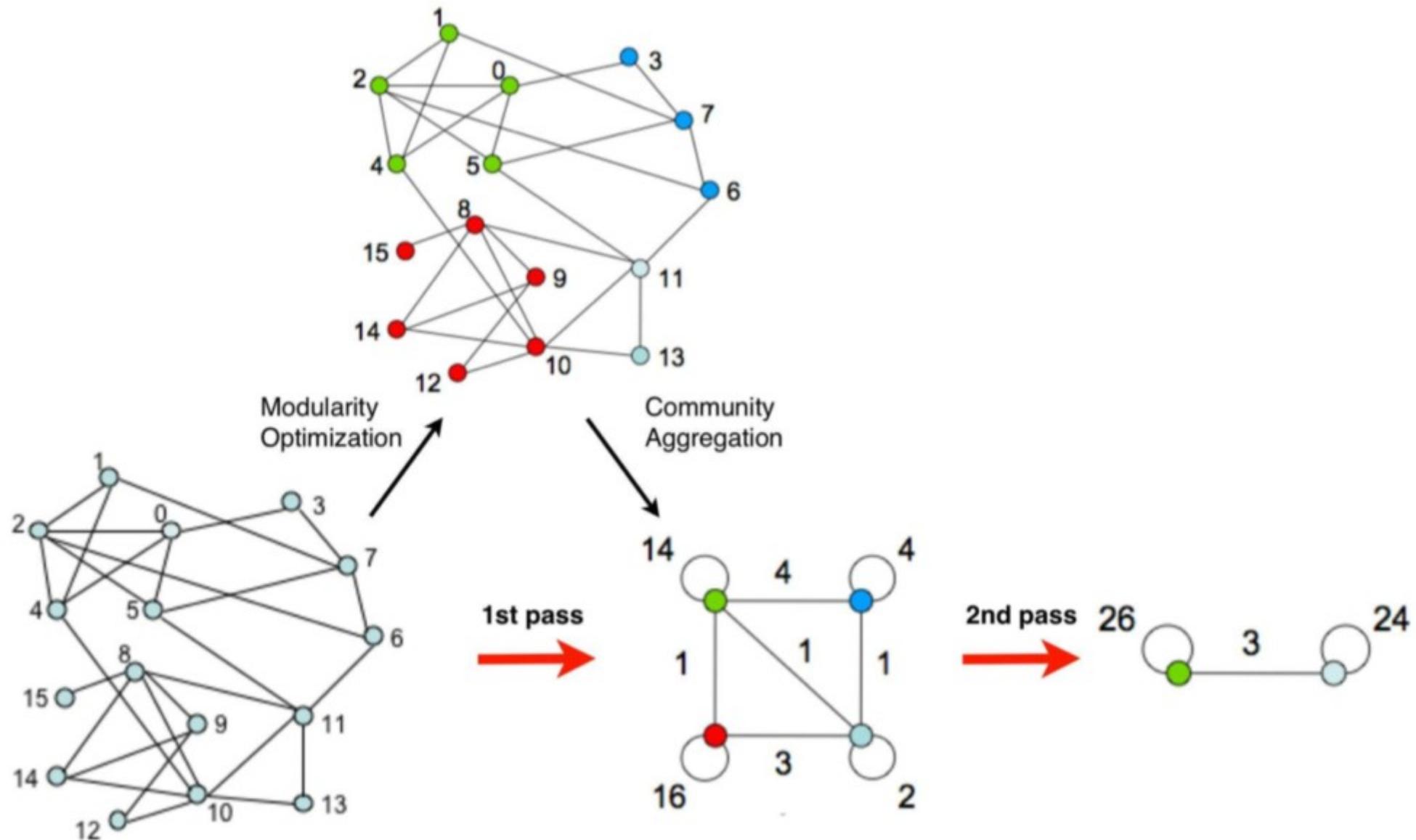
- where:
 - Σ_{in} ... sum of link weights between nodes in C
 - Σ_{tot} ... sum of all link weights of nodes in C
 - $k_{i,in}$... sum of link weights between node i and C
 - k_i ... sum of all link weights (i.e., degree) of node i
- Also need to derive $\Delta Q(D \rightarrow i)$ of taking node i out of community D .
- And then: $\Delta Q = \Delta Q(i \rightarrow C) + \Delta Q(D \rightarrow i)$



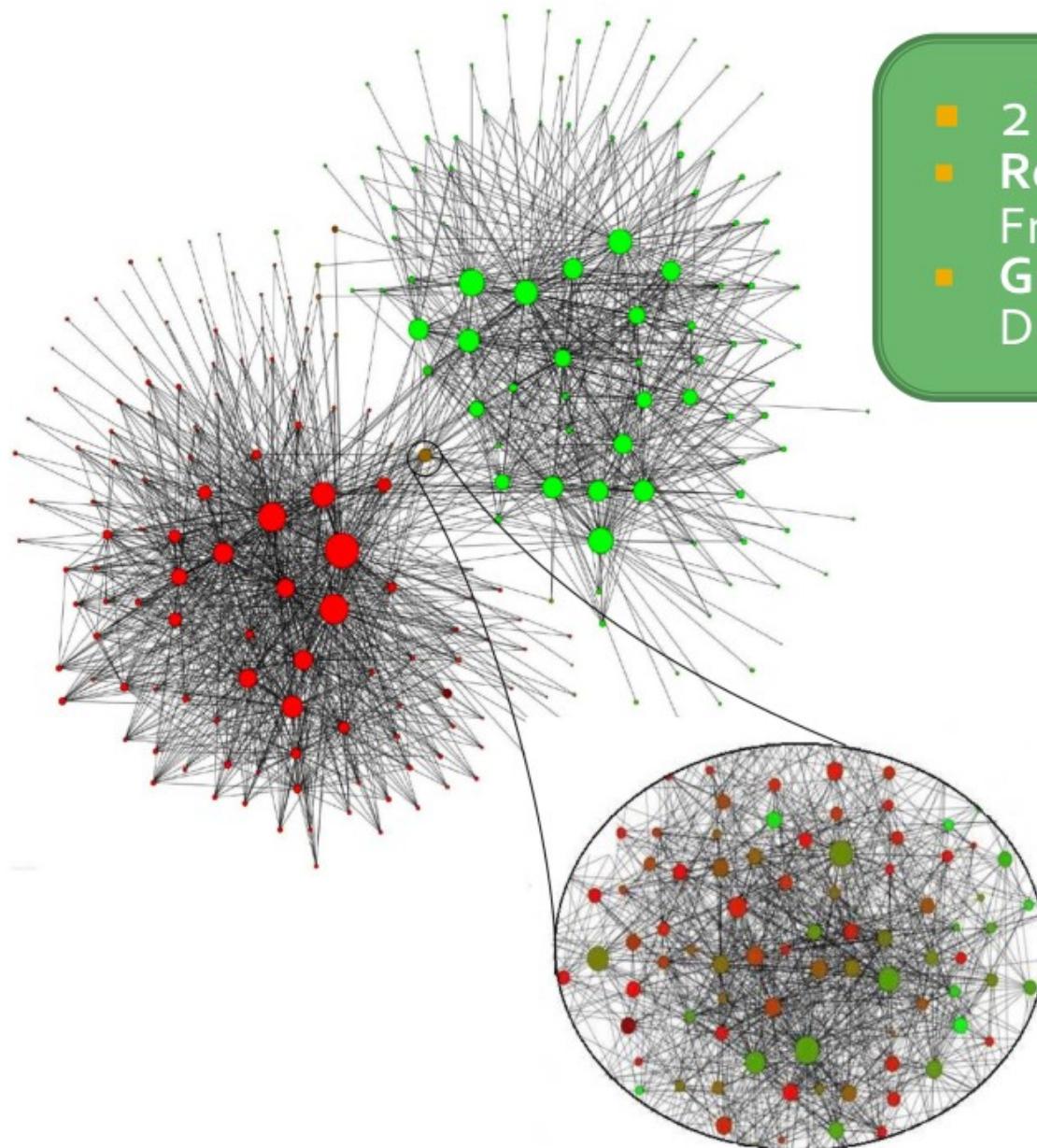
Louvain: 2nd phase (restructuring)

- The partitions obtained in the first phase are contracted into **super-nodes**, and the network is created accordingly
 - Super-nodes are connected if there is at least one edge between nodes of the corresponding partitions
 - The weight of the edge between the two super-nodes is the sum of the weights from all edges between their corresponding partitions
- **The loop runs until the community configuration does not change anymore**

Louvain Algorithm Overview



Loouvain: Belgian Phone Network



- 2M nodes
- Red nodes: French speakers
- Green nodes: Dutch speakers

There are many algorithms

Community detection in graphs

Santo Fortunato*

Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY.

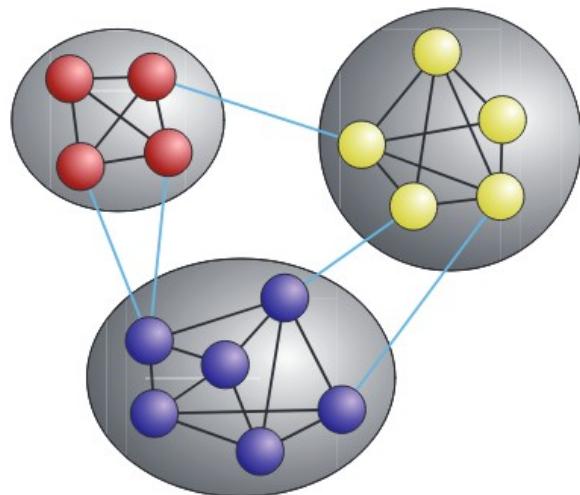
Contents	
I. Introduction	2
II. Communities in real-world networks	4
III. Elements of Community Detection	8
A. Computational complexity	9
B. Communities	9
1. Basics	9
2. Local definitions	10
3. Global definitions	11
4. Definitions based on vertex similarity	12
C. Partitions	13
1. Basics	13
2. Quality functions: modularity	14
IV. Traditional methods	16
A. Graph partitioning	16
B. Hierarchical clustering	19
C. Partitional clustering	19
D. Spectral clustering	20
V. Divisive algorithms	23
A. The algorithm of Girvan and Newman	23
B. Other methods	25
VI. Modularity-based methods	27
A. Modularity optimization	27
1. Greedy techniques	27
2. Simulated annealing	29
3. Extremal optimization	29
4. Spectral optimization	30
5. Other optimization strategies	33
B. Modifications of modularity	34
C. Limits of modularity	38
VII. Spectral Algorithms	41
VIII. Dynamic Algorithms	43
A. Spin models	43
IX. Methods based on statistical inference	48
A. Generative models	49
B. Blockmodeling, model selection and information theory	52
X. Alternative methods	54
XI. Methods to find overlapping communities	58
A. Clique percolation	58
B. Other techniques	60
XII. Multiresolution methods and cluster hierarchy	62
A. Multiresolution methods	63
B. Hierarchical methods	65
XIII. Detection of dynamic communities	66
XIV. Significance of clustering	70
XV. Testing Algorithms	73
A. Benchmarks	74
B. Comparing partitions: measures	77
C. Comparing algorithms	79
XVI. General properties of real clusters	82
XVII. Applications on real-world networks	85
A. Biological networks	85
B. Social networks	86
C. Other networks	88
XVIII. Outlook	90
A. Elements of Graph Theory	92
1. Basic Definitions	92
2. Graph Matrices	94
3. Model graphs	94
References	96

*Electronic address: fortunato@isi.it

There are many algorithms

Community detection in networks: A user guide

Santo Fortunato*



Contents

I. Introduction	1
II. What are communities?	3
A. Variables	3
B. Classic view	5
C. Modern view	7
III. Validation	9
A. Artificial benchmarks	9
B. Partition similarity measures	12
C. Detectability	15
D. Structure versus metadata	17
E. Community structure in real networks	19
IV. Methods	22
A. How many clusters?	22
B. Consensus clustering	24
C. Spectral methods	25
D. Overlapping communities: Vertex or Edge clustering?	25
E. Methods based on statistical inference	27
F. Methods based on optimisation	27
G. Methods based on dynamics	31
H. Dynamic clustering	34
I. Significance	35
J. Which method then?	37
V. Software	38
VI. Outlook	39
Acknowledgments	40
References	40

There are many “score” functions

Defining and Evaluating Network Communities based on Ground-truth

Jaewon Yang
Stanford University
crucis@stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

Dataset	<i>N</i>	<i>E</i>	<i>C</i>	<i>S</i>	<i>A</i>
LiveJournal	4.0M	34.9M	311,782	40.06	3.09
Friendster	117.7M	2,586.1M	1,449,666	26.72	0.32
Orkut	3.0M	117.2M	8,455,253	34.86	95.9
Ning (225 nets)	7.0M	35.5M	137,177	46.89	0.92
Amazon	0.33M	0.92M	49,732	99.86	14.83
DBLP	0.42M	1.34M	2,547	429.79	2.56

Table I

230 SOCIAL, COLLABORATION AND INFORMATION NETWORKS WITH EXPLICIT GROUND-TRUTH COMMUNITIES. *N*: NUMBER OF NODES, *E*: NUMBER OF EDGES, *C*: NUMBER OF COMMUNITIES, *S*: AVERAGE COMMUNITY SIZE, *A*: COMMUNITY MEMBERSHIPS PER NODE. NING STATISTICS ARE AGGREGATED OVER 225 DIFFERENT SUBNETWORKS.

(A) Scoring functions based on internal connectivity:

- **Internal density:** $f(S) = \frac{m_S}{n_S(n_S-1)/2}$ is the internal edge density of the node set S [24].
- **Edges inside:** $f(S) = m_S$ is the number of edges between the members of S [24].
- **Average degree:** $f(S) = \frac{2m_S}{n_S}$ is the average internal degree of the members of S [24].
- **Fraction over median degree (FOMD):**

$$f(S) = \frac{|\{u:u \in S, \{\{(u,v):v \in S\} > d_m\}|}{n_S}$$
is the fraction of nodes of S that have internal degree higher than d_m , where d_m is the median value of $d(u)$ in V .
- **Triangle Participation Ratio (TPR):**

$$f(S) = \frac{|\{u:u \in S, \{(v,w):v \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}|}{n_S}$$
is the fraction of nodes in S that belong to a triad.

(B) Scoring functions based on external connectivity:

- **Expansion** measures the number of edges per node that point outside the cluster: $f(S) = \frac{e_S}{n_S}$ [24].
- **Cut Ratio** is the fraction of existing edges (out of all possible edges) leaving the cluster: $f(S) = \frac{e_S}{n_S(n-n_S)}$ [9].

(C) Scoring functions that combine internal and external connectivity:

- **Conductance:** $f(S) = \frac{e_S}{2m_S + e_S}$ measures the fraction of total edge volume that points outside the cluster [27].
- **Normalized Cut:** $f(S) = \frac{e_S}{2m_S + e_S} + \frac{e_S}{2(m-n_S) + e_S}$ [27].
- **Maximum-ODF (Out Degree Fraction):**

$$f(S) = \max_{u \in S} \frac{|\{(u,v) \in E: v \notin S\}|}{d(u)}$$
is the maximum fraction of edges of a node in S that point outside S [8].
- **Average-ODF:** $f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v) \in E: v \notin S\}|}{d(u)}$ is the average fraction of edges of nodes in S that point out of S [8].
- **Flake-ODF:** $f(S) = \frac{|\{u:u \in S, |\{(u,v) \in E: v \in S\}| < d(u)/2\}|}{n_S}$ is the fraction of nodes in S that have fewer edges pointing inside than to the outside of the cluster [8].

(D) Scoring function based on a network model:

- **Modularity:** $f(S) = \frac{1}{4}(m_S - E(m_S))$ is the difference between m_S , the number of edges between nodes in S and $E(m_S)$, the expected number of such edges in a random graph with identical degree sequence [21].

War Story

FastStep: Scalable Boolean Matrix Decomposition

Miguel Araujo^{1,2}, Pedro Ribeiro¹, and Christos Faloutsos²

¹ Cracs/INESC-TEC and University of Porto, Porto, Portugal
pribheiro@dcc.fc.up.pt

² Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
{maraaujo, christos}@cs.cmu.edu

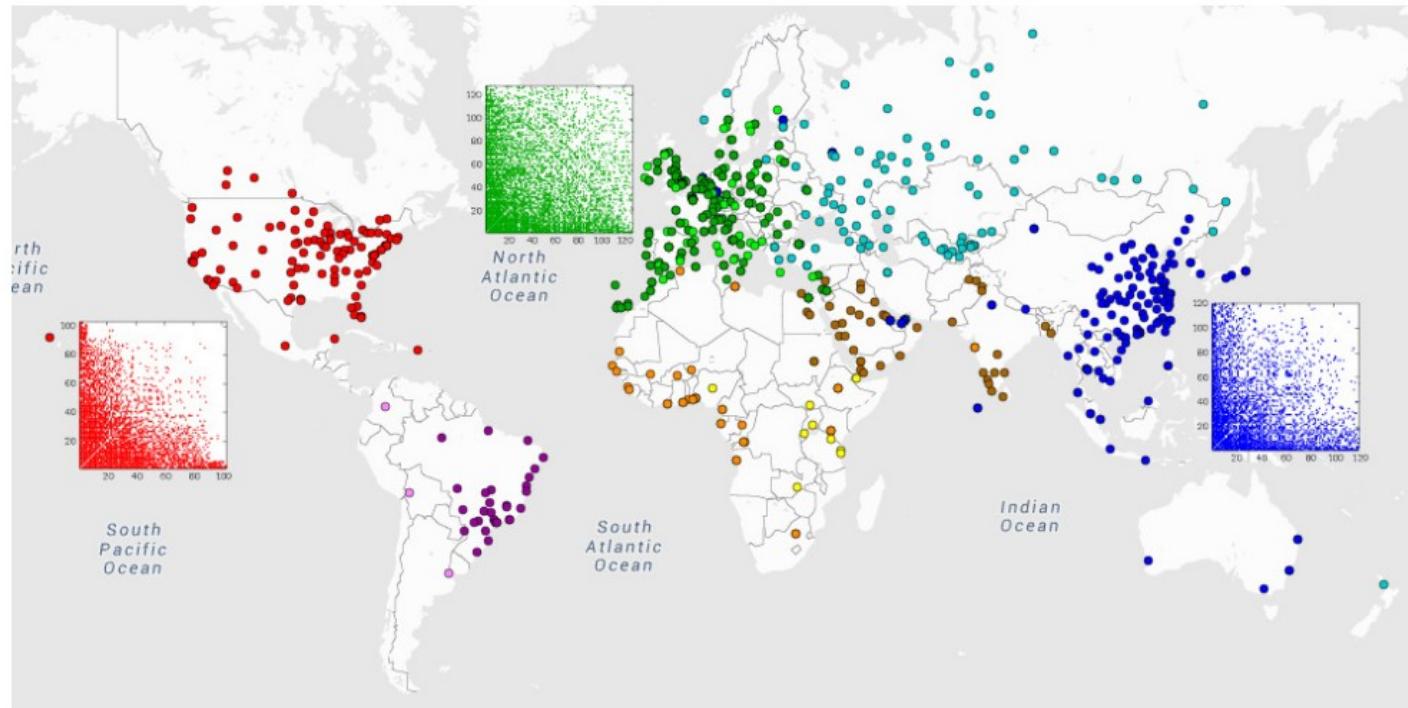
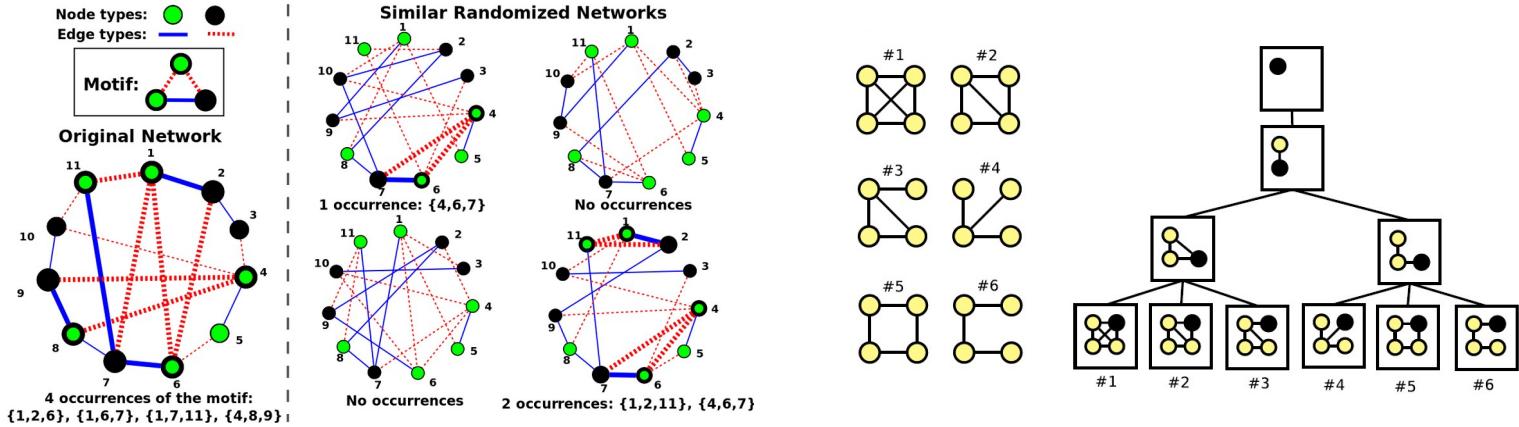


Table 1: Comparison of decomposition models based on interpretability and beyond block structure

	FASTSTEP	SVD	T
Scalability	✓	✓	
Overlapping	✓	✓	
Beyond blocks	✓	✓	✓
Boolean Reconstruction	✓		✓
Arbitrary Marginals	✓	✓	✓
Interpretability	✓		✓



Subgraphs as Fundamental Ingredients of Complex Networks

Concepts, Methods and Applications

Contents

1) Motivation

2) Concepts

3) Computational Challenge
(and sequential exact solutions)

4) Sampling approach

5) Parallel Approach

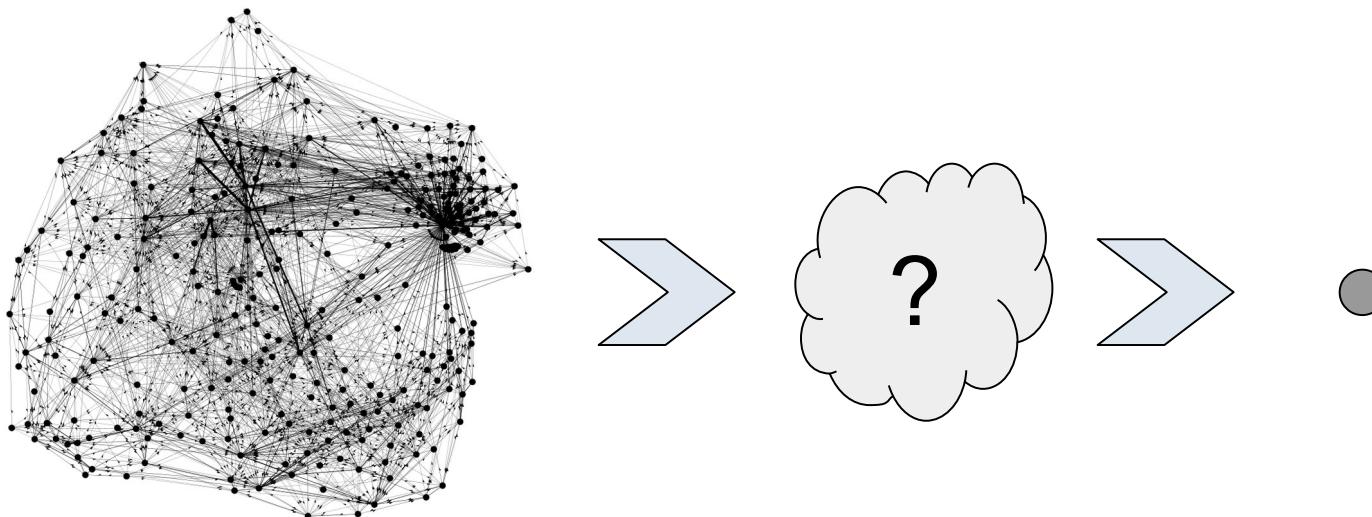
6) Example Applications

7) Resources

1) MOTIVATION

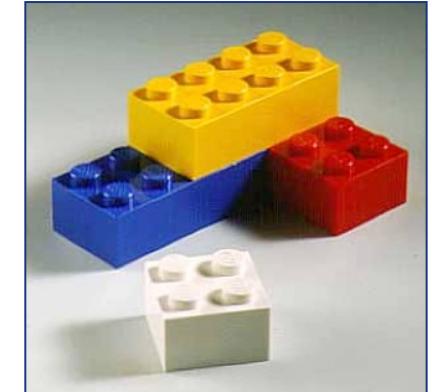
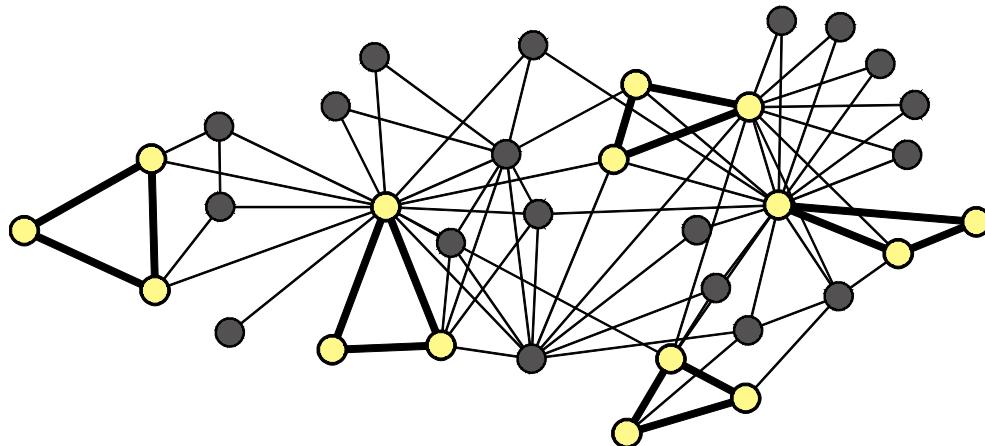
Network Metrics

- **There are many available metrics at the node level:**
 - E.g. degree, betweenness, closeness
- **There are also many metrics at the global level:**
 - E.g. diameter, avg. distance, density, clustering coefficient
- **What about something inbetween?**



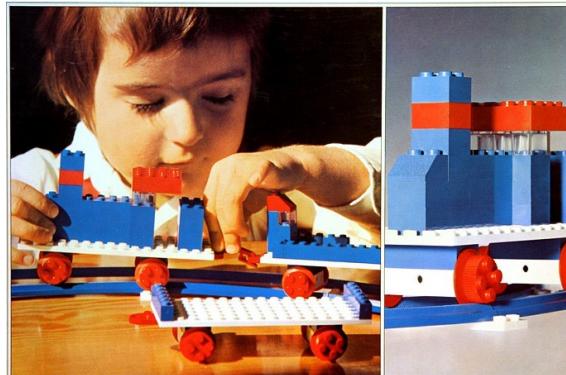
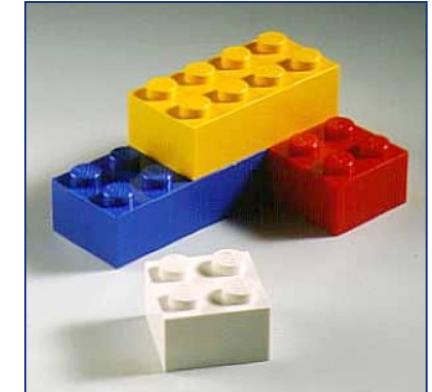
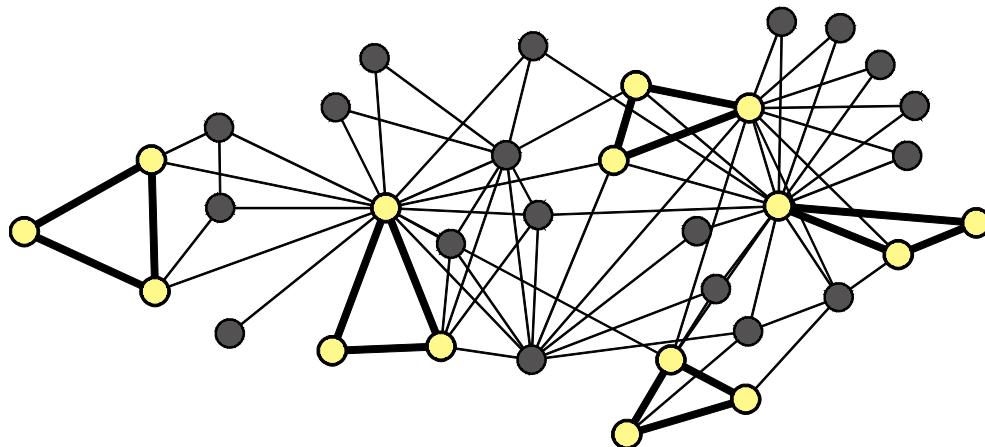
Building Blocks of Networks

- **Subnetworks, or subgraphs, are the building blocks of networks**



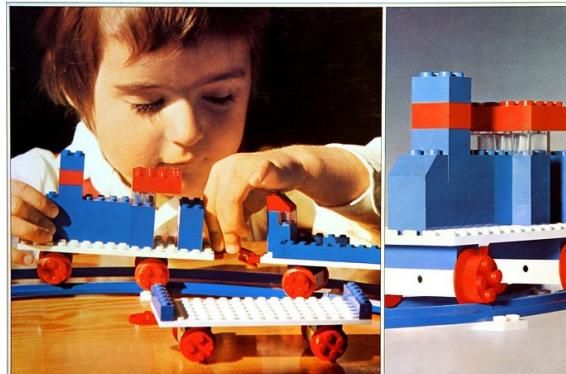
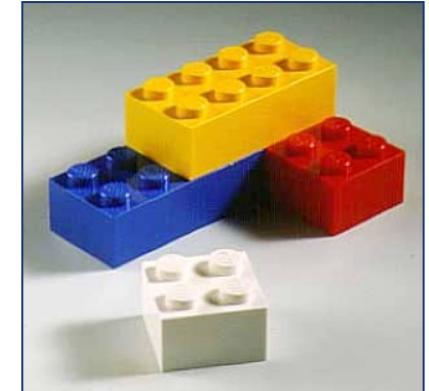
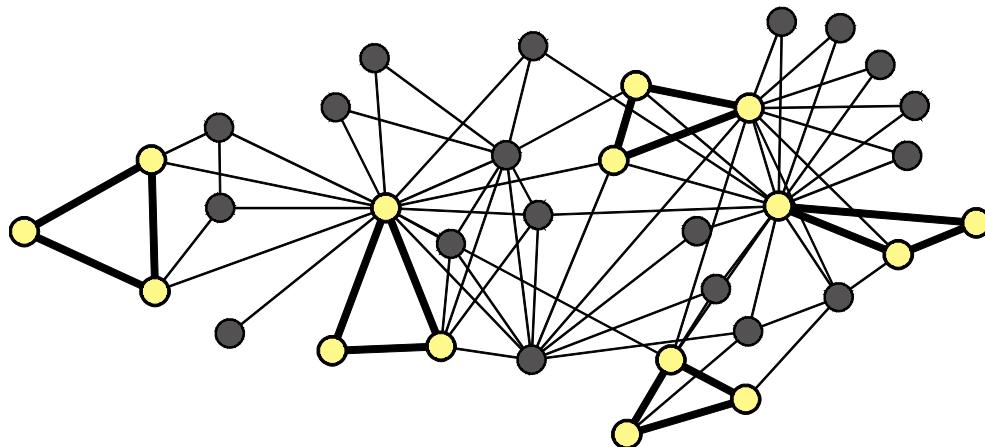
Building Blocks of Networks

- **Subnetworks, or subgraphs, are the building blocks of networks**



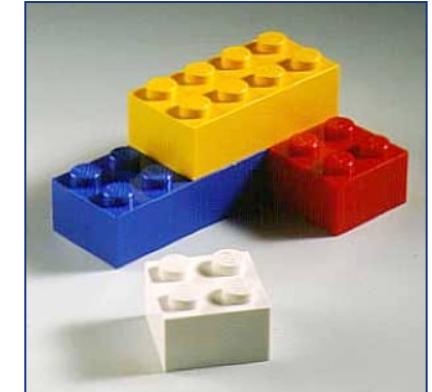
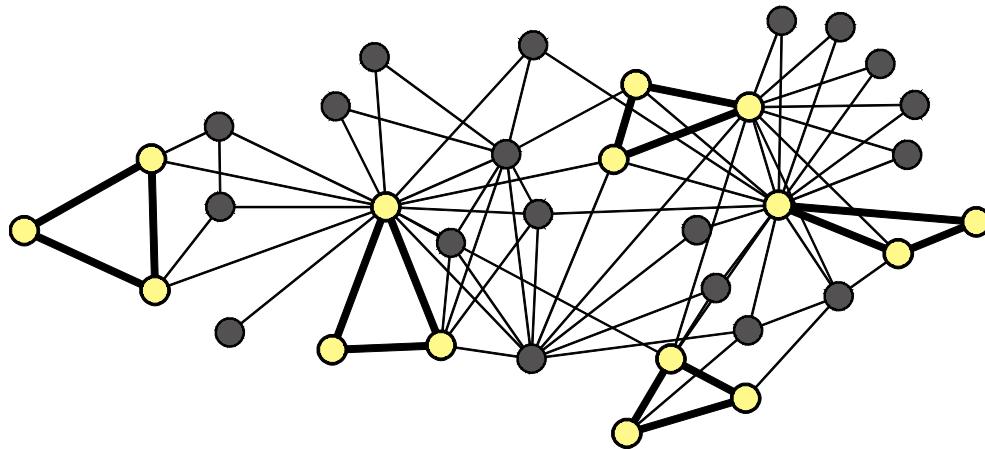
Building Blocks of Networks

- **Subnetworks, or subgraphs, are the building blocks of networks**



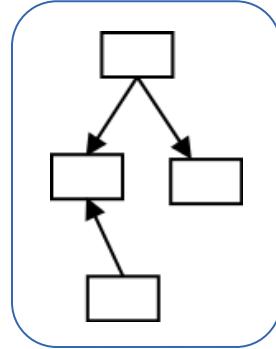
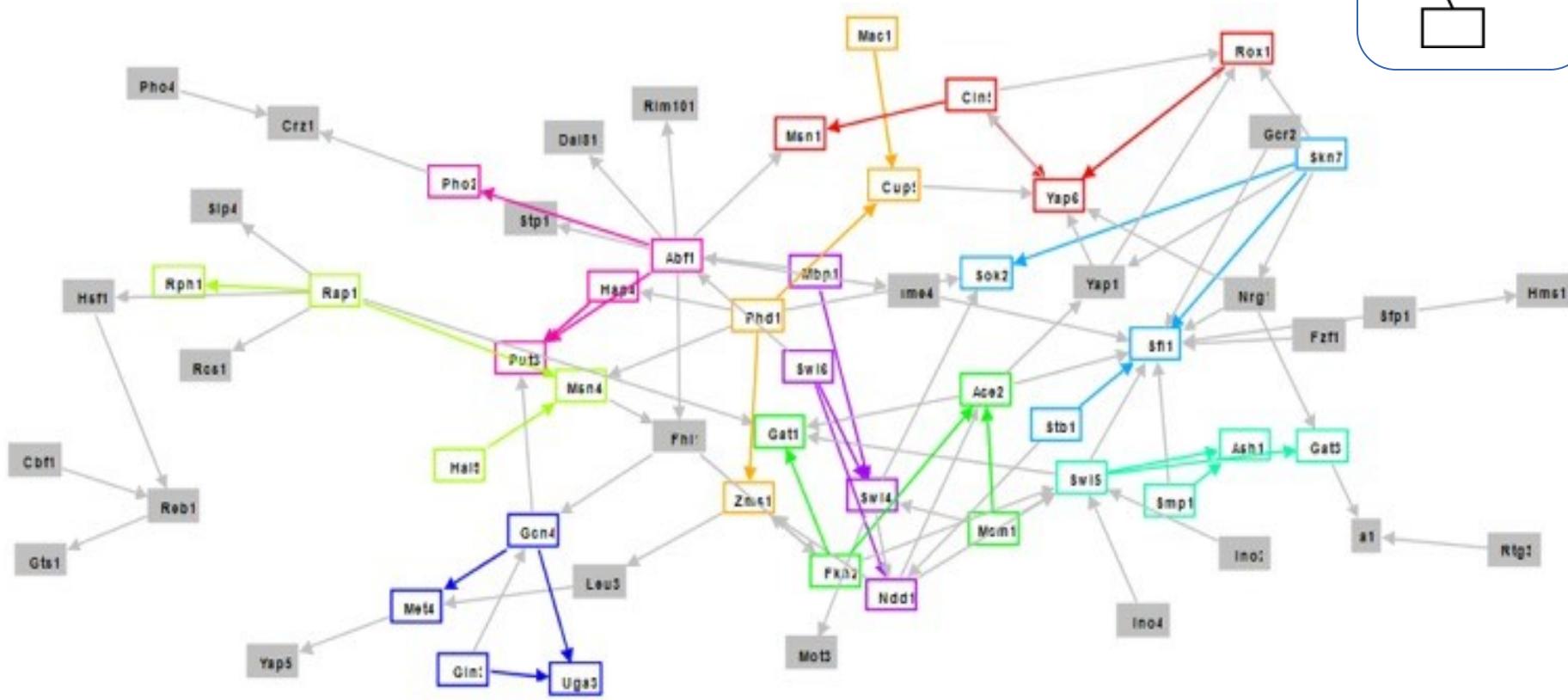
Building Blocks of Networks

- **Subnetworks, or subgraphs, are the building blocks of networks**



- **They have the power to characterize and discriminate networks**

Building Blocks of Networks

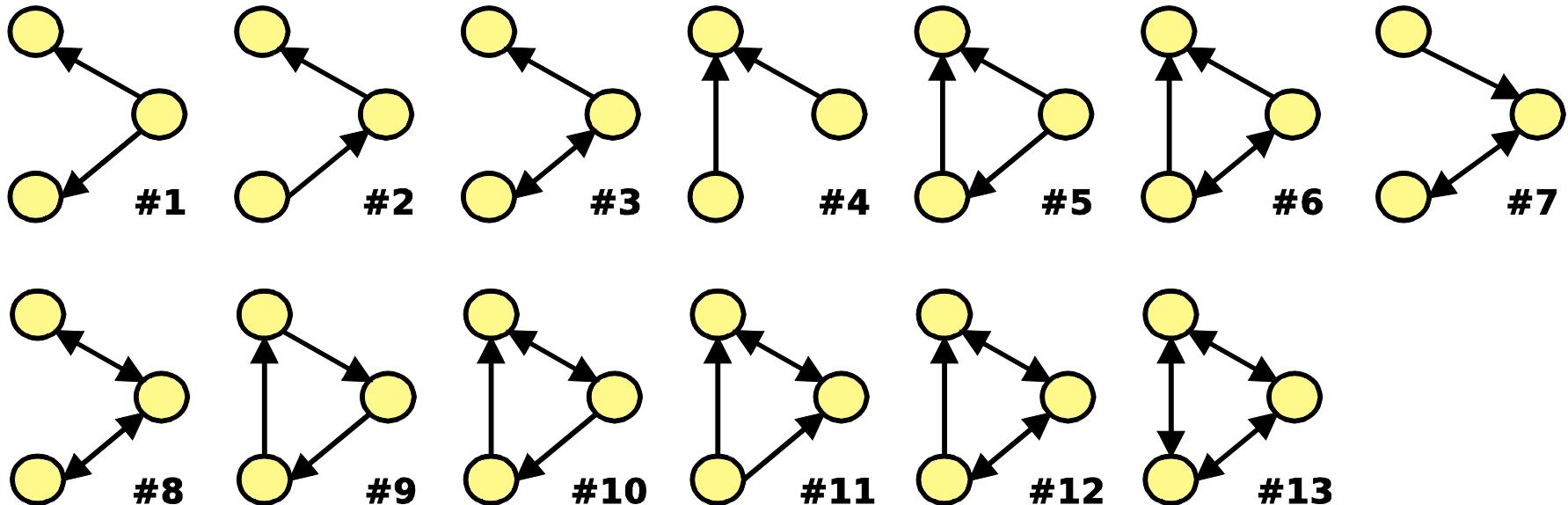


Subgraphs: fundamental ingredients of networks

Pedro Ribeiro

Example Application

- Consider all possible directed subgraphs of size 3



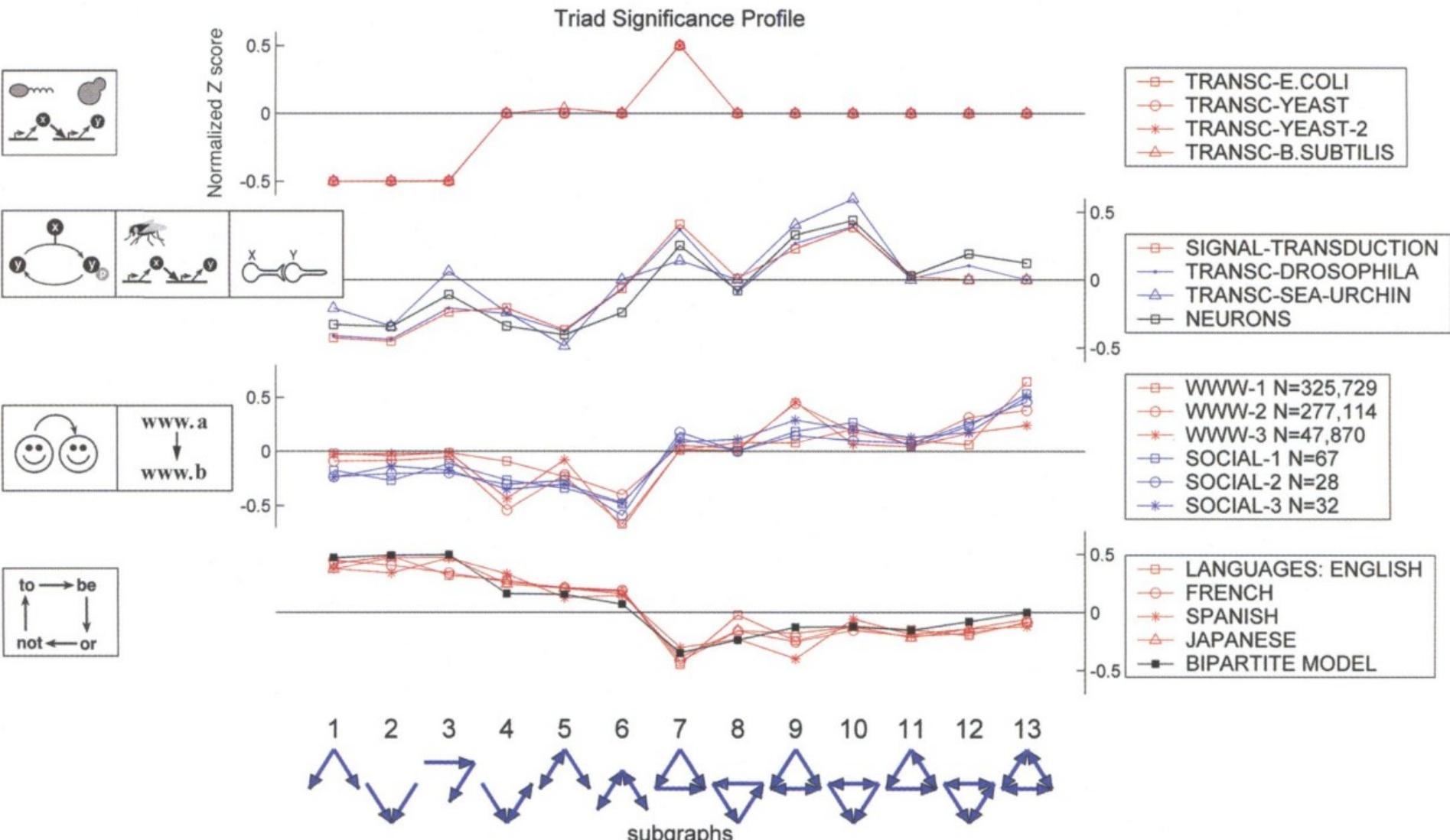
Example Application

- **For each subgraph type:**
 - Metric capable of classifying subgraph “significance”
[more about that later]
 - Values in interval [-1,1]
 - **Negative** values indicate **underepresentation**
 - **Positive** values indicate **overrepresentation**
- **With this you could create a network fingerprint:**
 - Feature vector with all subgraph significances

Example Application

- Consider the following varied types of networks:
 - Regulatory Network (gene regulation)
 - Neuronal Network (synaptic connections)
 - World Wide Web (hyperlinks between pages)
 - Social network (friendships)
 - Semantic Networks (word adjacency)
- What happens when we look at their fingerprints as defined before?

Example Application

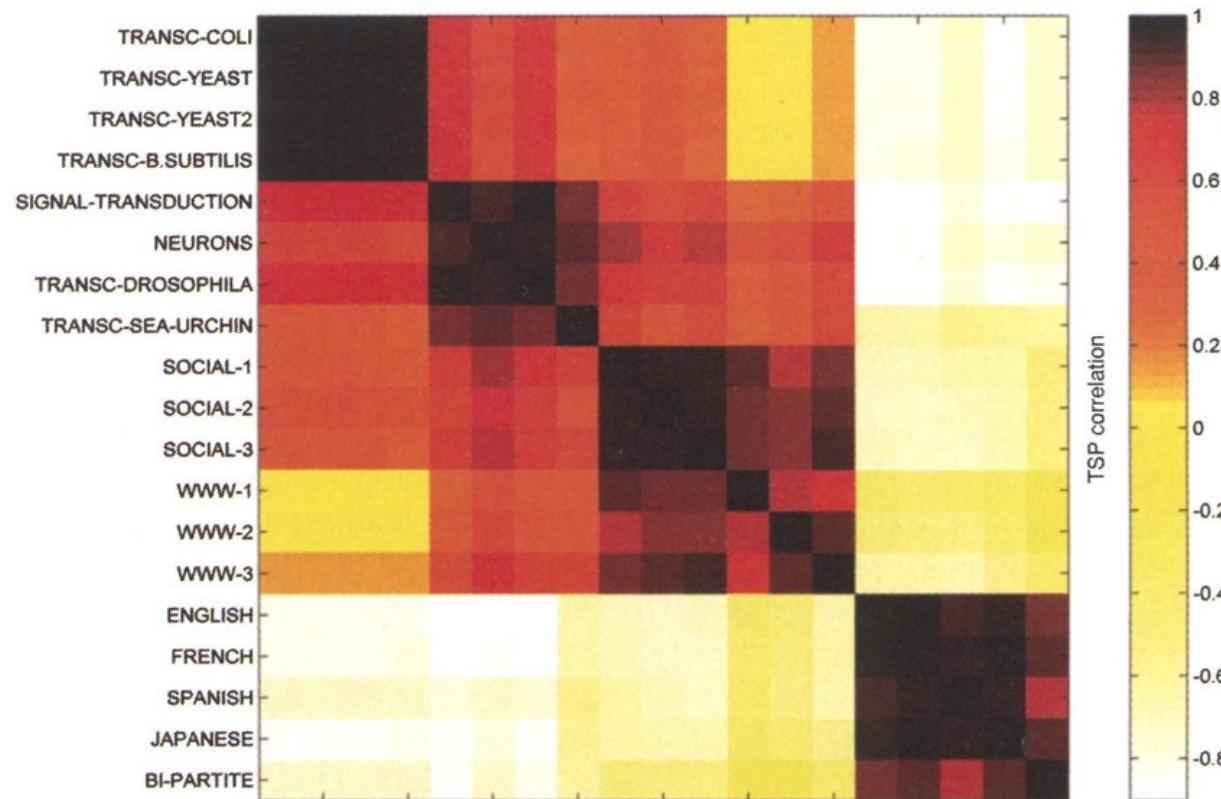


Different networks have similar fingerprints!

Image: (Milo et al., 2004)

Example Application

Correlation



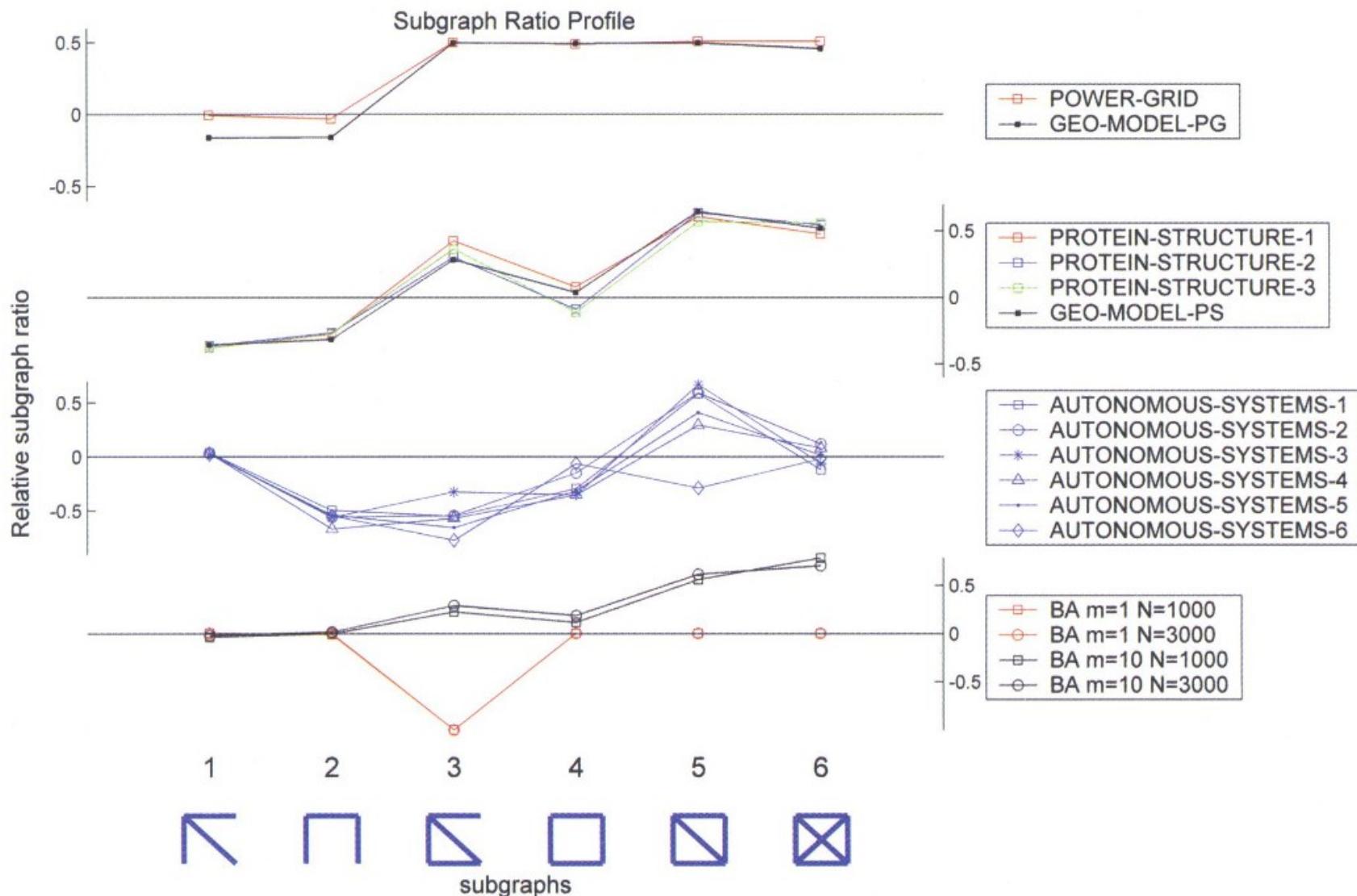
Different networks have similar fingerprints!

Image: (Milo et al., 2004)

Example Application

- **What about undirected networks?**
- **Consider the following types of networks:**
 - Power Grid (electrical geographical power grid)
 - Protein Structure (secondary structure adjacency)
 - Autonomous Systems (internet)
- **What happens when we look at their fingerprints as defined before?**

Example Application



Different networks have similar fingerprints!

Image: (Milo et al., 2004)

Subgraphs are powerful

**Subgraphs have the power to
characterize and discriminate
networks**

Their applicability is general

2) CONCEPTS

Network Motifs

- **Milo et al. (2002) came up with the definition of network motifs:**
 - “recurring, significant patterns of interconnections”
- **How to define:**
 - **Pattern:** induced subgraph
 - **Recurring:** found many times, i.e., high frequency
 - **Significant:** more frequent than it would be expected in similar networks (same degree sequence)

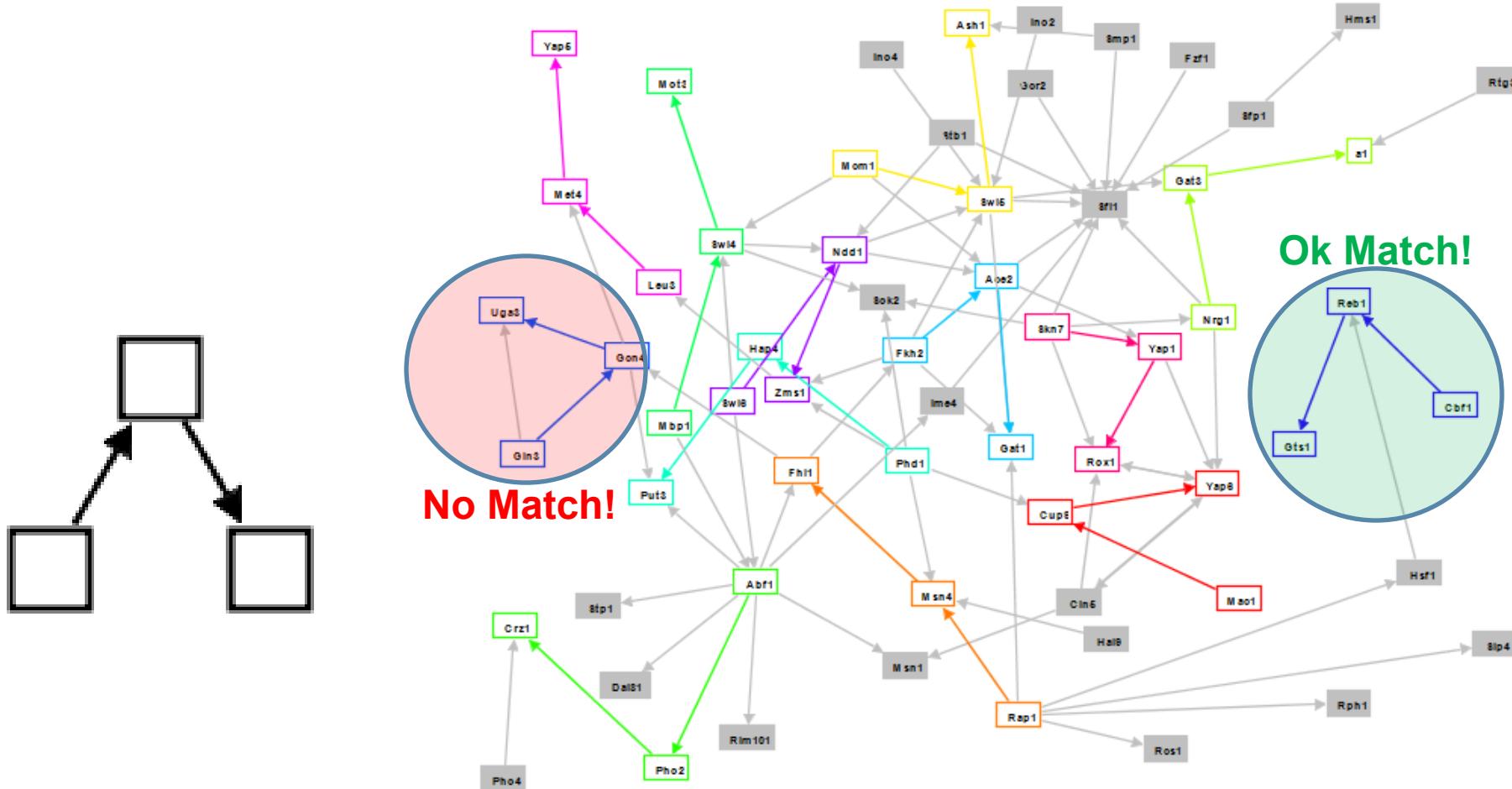
- 1) $\text{Prob}(\bar{f}_{\text{random}}(G_K) > f_{\text{original}}(G_K)) \leq P$
(Over-representation)
- 2) $f_{\text{original}}(G_K) \geq U$
(Minimum frequency)
- 3) $f_{\text{original}}(G_K) - \bar{f}_{\text{random}}(G_K) > D \times \bar{f}_{\text{random}}(G_K)$
(Minimum deviation)

Parameters P, U, D, N
control the definition
(Milo et al., 2002, used
 $\{0.01, 4, 0.1, 1000\}$)

Image: Adapted from (Milo et al., 2004)

Subgraph concepts - Induced

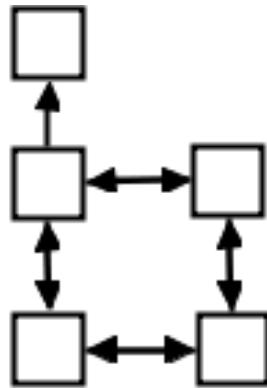
Induced Subgraphs



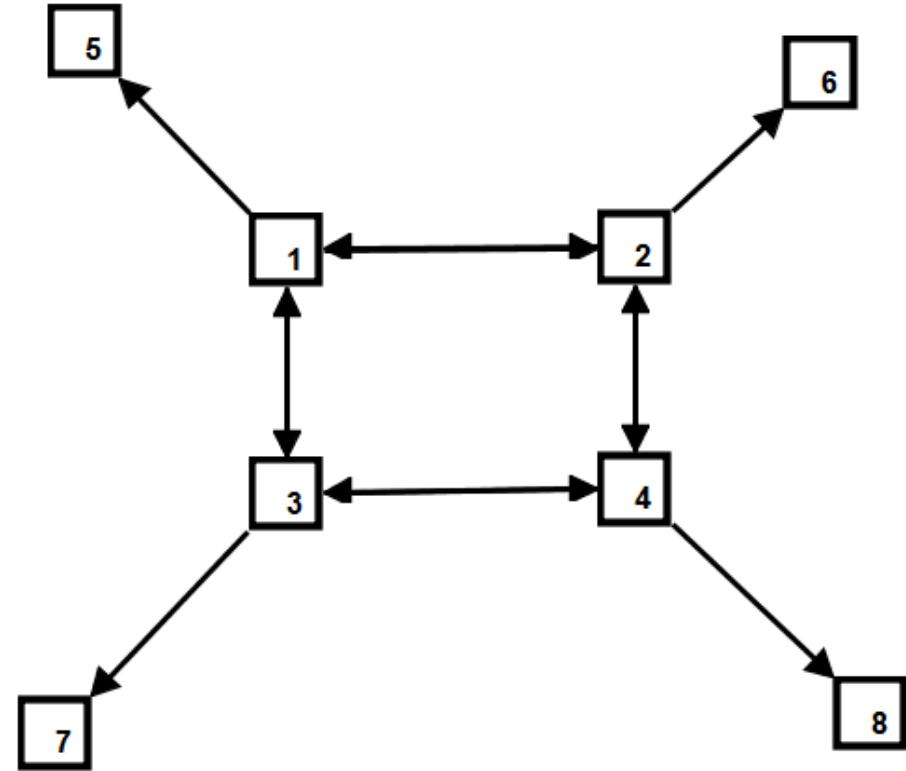
Subgraph concepts - Frequency

- How to count?

- Allow **overlapping**



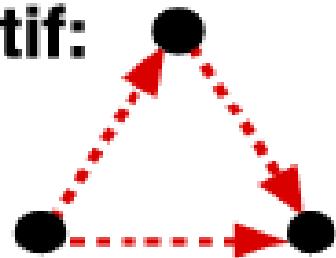
- 4 occurrences:
 $\{1,2,3,4,5\}$
 $\{1,2,3,4,6\}$
 $\{1,2,3,4,7\}$
 $\{1,2,3,4,8\}$



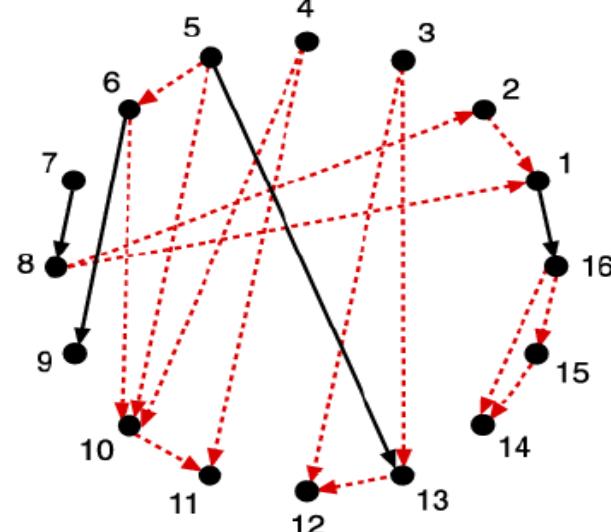
Subgraph concepts – Significance

Traditional Null Model – keep **Degree Sequence**

motif:



Motif



Original Network

Random Networks

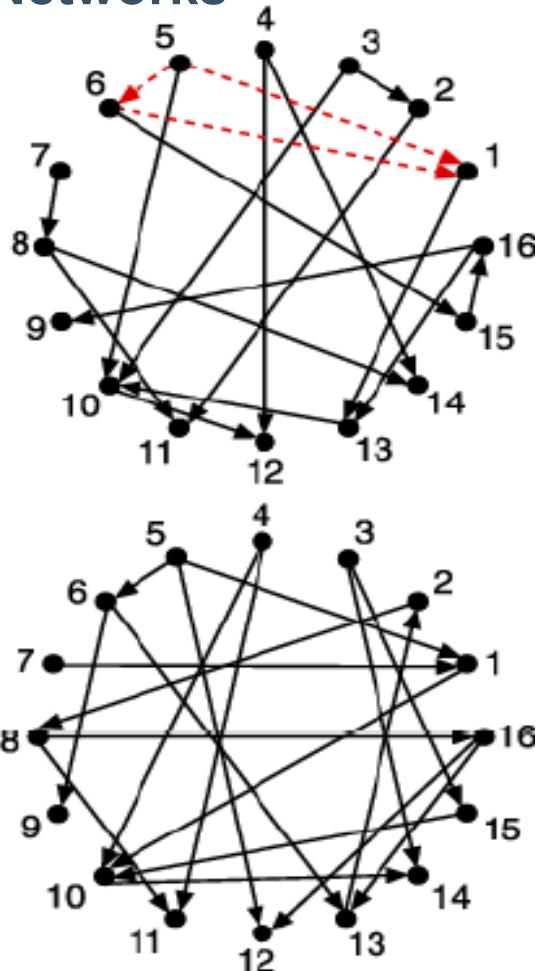
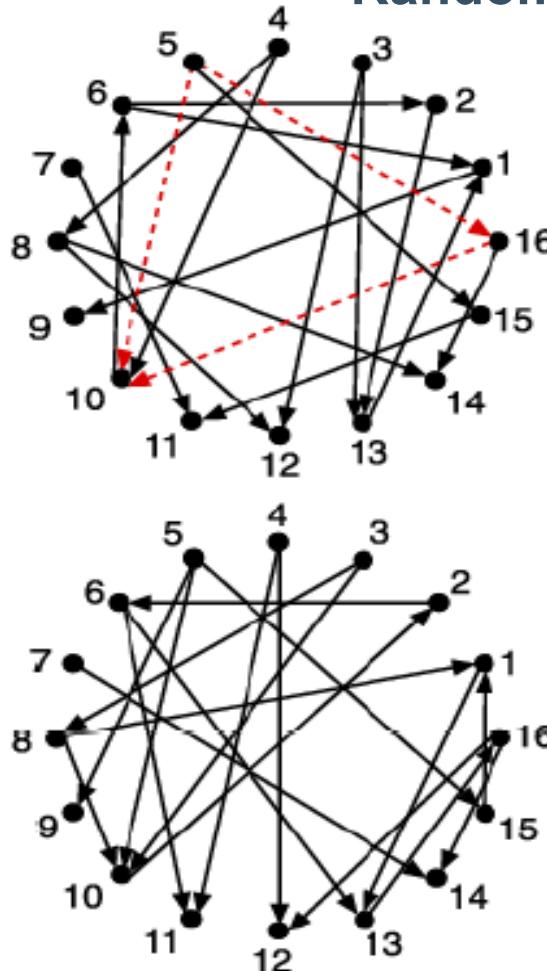
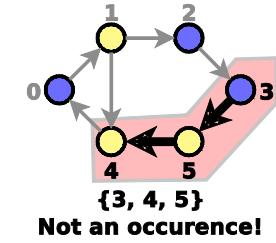
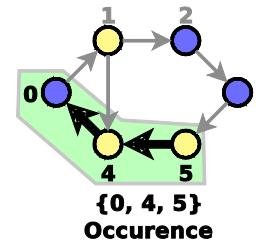
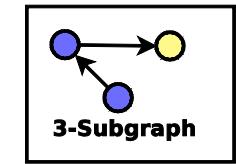
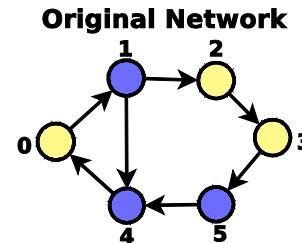


Image: Adapted from (Milo et al., 2002)

Network Motifs Applicability

- **Canon definition:**

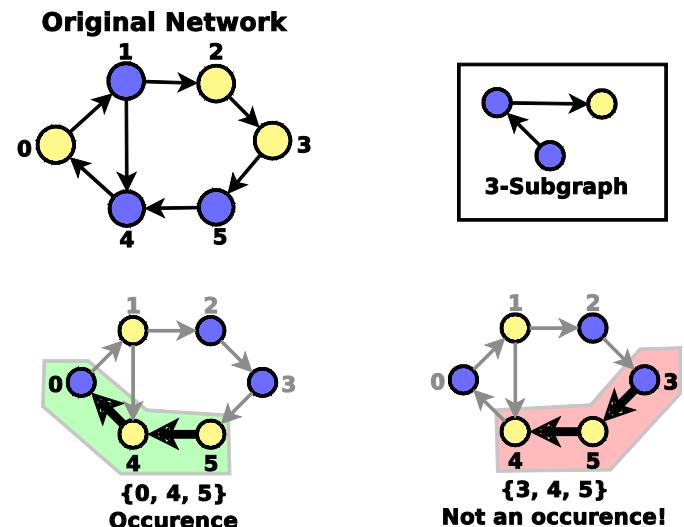
- Directed and Undirected
- Colored and uncolored



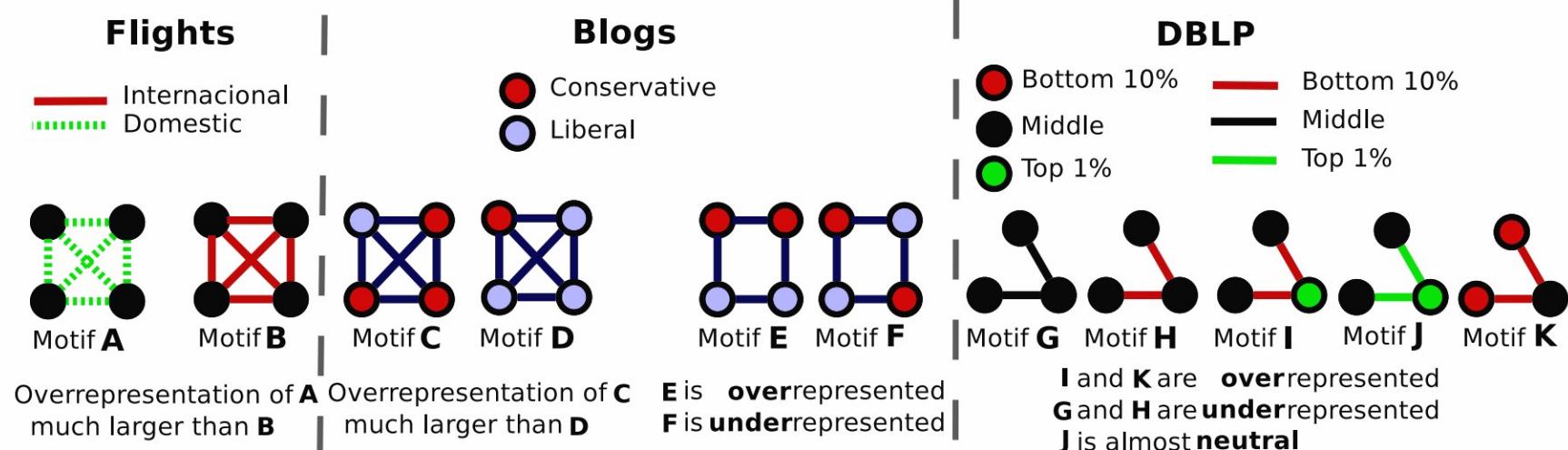
Network Motifs Applicability

• Canon definition:

- Directed and Undirected
- Colored and uncolored



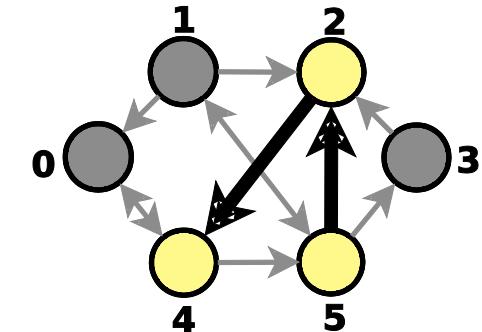
Example application of colored motifs: [Ribeiro & Silva, Complenet'2014]



Network Motifs Applicability

• Variations on the concept

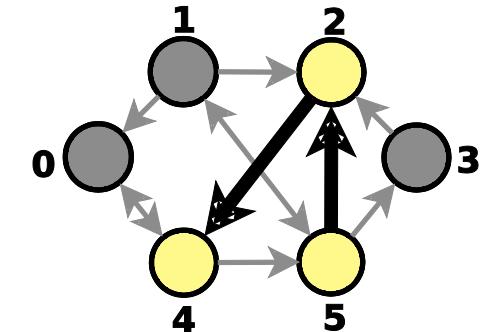
- Different frequency concepts
- Different significance metrics
- Under-Representation (**anti-motifs**)
- Weighted networks
- Different constraints for the null model



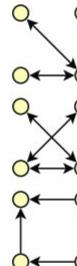
Network Motifs Applicability

• Variations on the concept

- Different frequency concepts
- Different significance metrics
- Under-Representation (**anti-motifs**)
- Weighted networks
- Different constraints for the null model



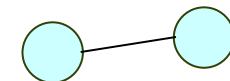
Ex. application of different null model: [Silva, Paredes & Ribeiro, Complenet'2017]

Network	K	Subgraph	Original	Keep $K - 1$		
				Keep Deg. Seq.	Change Deg. Seq.	ER
Macaque Cortex	4		61.20 ^a	-2.29	-0.71	-4.41
			182.30 ^a	6.19	2.47	12.66
			-10.17 ^b	12.01	10.64	15.20

Random networks with prescribed degree frequencies

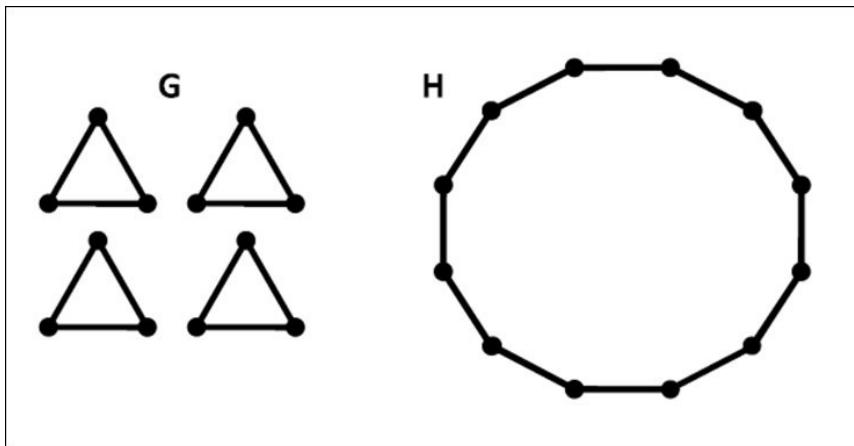
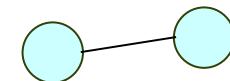
Graphlet Degree Distribution

- What about a “node-level” subgraph metric?
- The degree distribution is in a way measuring participation in subgraphs of size 2
 - Can we generalize this?



Graphlet Degree Distribution

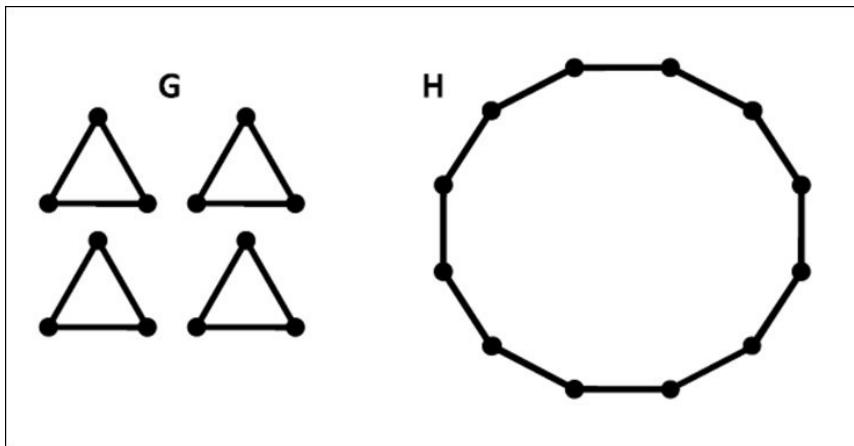
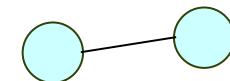
- What about a “node-level” subgraph metric?
- The degree distribution is in a way measuring participation in subgraphs of size 2
 - Can we generalize this?



The same degree distribution can correspond to very different networks!

Graphlet Degree Distribution

- What about a “node-level” subgraph metric?
- The degree distribution is in a way measuring participation in subgraphs of size 2
 - Can we generalize this?

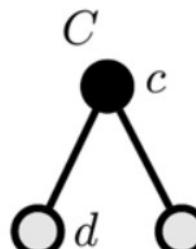
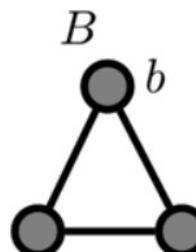
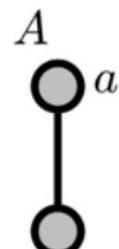
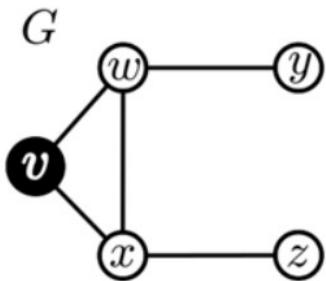


The same degree distribution can correspond to very different networks!

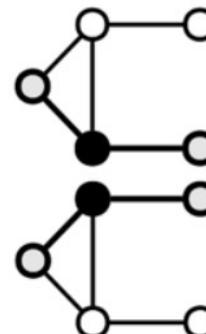
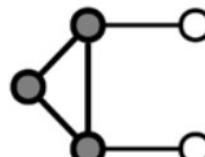
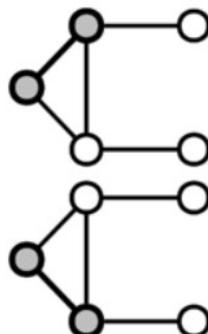
- Przulj (2006) came up with the definition of graphlet degree distribution:
 - Where does the node appear in orbits of subgraphs?

Graphlet Degree Vector

- An automorphism “orbit” takes into account the symmetries of the graph
- The graphlet degree vector (GDV) is a feature vector with the frequency of the node in each orbit position (GDD is the dgraphlet degree distribution)



orbit	a	b	c	d
$GDV(v)$	2	1	0	2



$$Fr_G = \begin{bmatrix} a & b & c & d \\ v & 2 & 1 & 0 & 2 \\ w & 3 & 1 & 2 & 1 \\ x & 3 & 1 & 2 & 1 \\ y & 1 & 0 & 2 & 0 \\ z & 1 & 0 & 2 & 0 \end{bmatrix}$$



$$GDD_G = \begin{bmatrix} 1 & 2 & 3 \\ a & 2 & 1 & 2 \\ b & 3 & 0 & 0 \\ c & 0 & 4 & 0 \\ d & 2 & 1 & 0 \end{bmatrix}$$

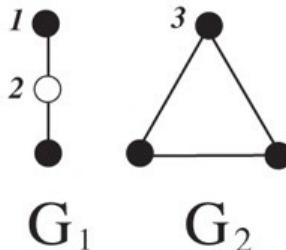
Graphlet Degree Distribution

Equivalent to “degree distribution”

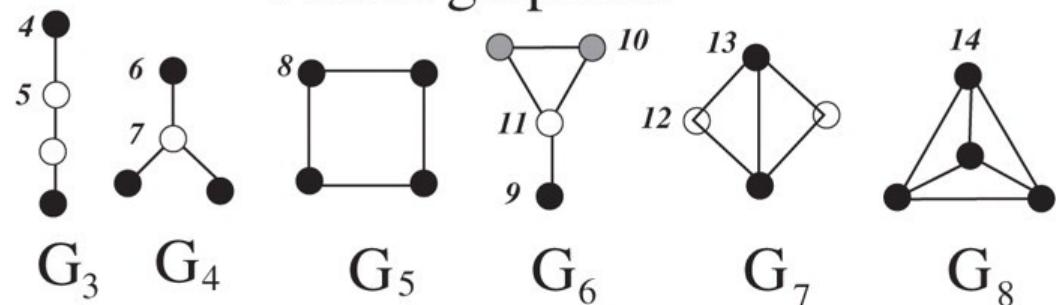
2-node graphlet



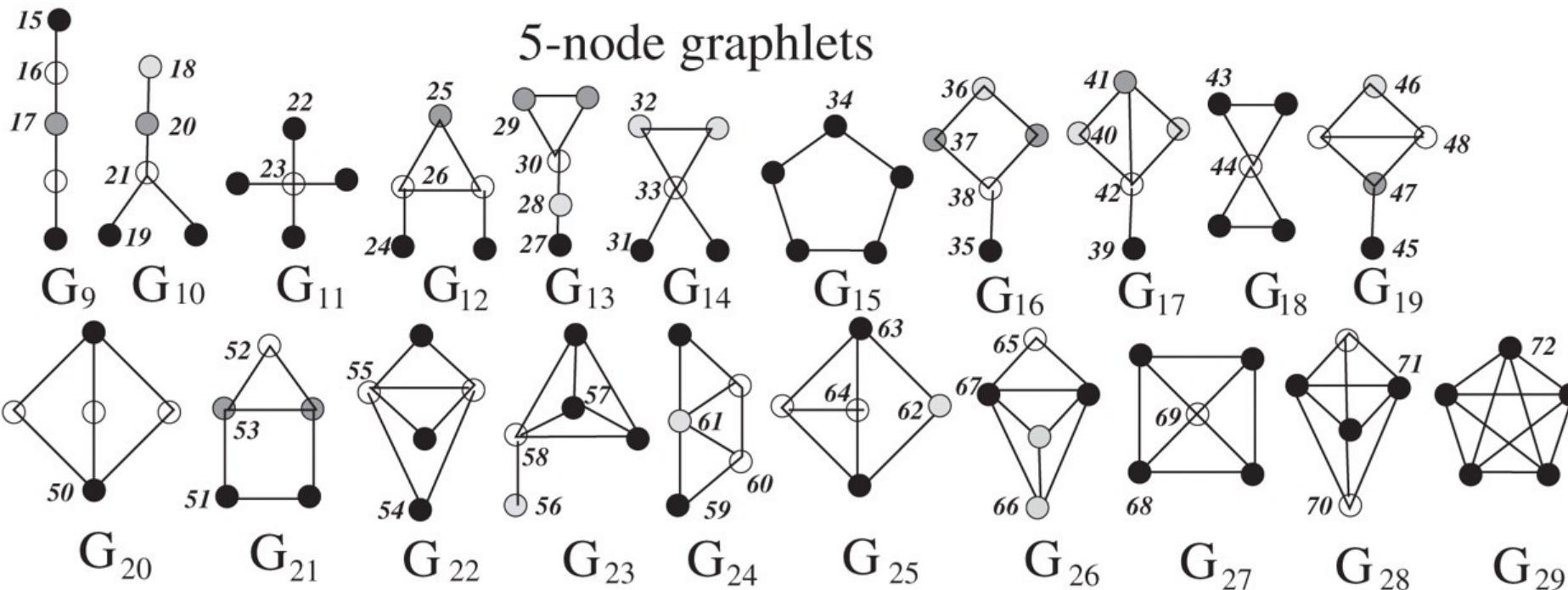
3-node graphlets



4-node graphlets



5-node graphlets



3) COMPUTATIONAL CHALLENGE

Computational Problem

- In its core, finding motifs and graphlets its all about **finding and counting subgraphs**.
- Just knowing if a certain subgraph exists is already an **hard computational problem!**
 - Subgraph isomorphism is NP-complete
- Execution time grows exponentially as the **size of the graph or the motif/graphlet increases**
 - Feasible motif size is usually small (3 to 8) and network size in the order of hundreds or thousands of nodes

What we have been doing

- Our primary goal was to improve efficiency in network motif detection.

Scale Up!

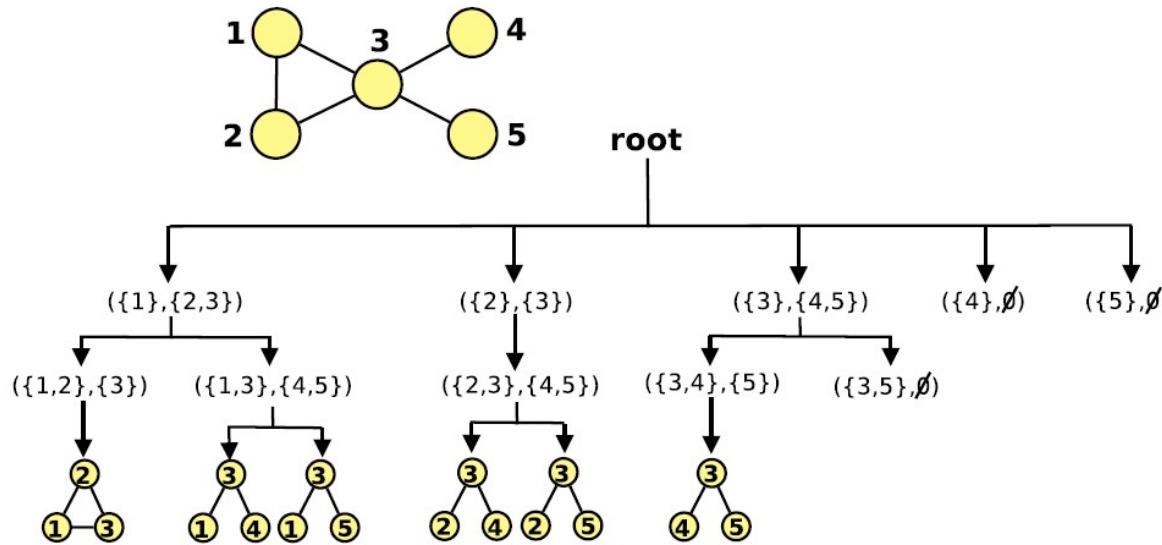


- How?
 - Novel **data structures** for the graphs and subgraphs
 - Novel faster **algorithms**
 - **Sampling** techniques
 - **Parallel** approaches (with different paradigms)

Previous Approaches

● **Network-centric approaches:**

- Enumerate all k -connected sets of nodes and then compute isomorphisms (ex: ESU/Fanmod, Kavosh)



• **Subgraph-centric approaches:**

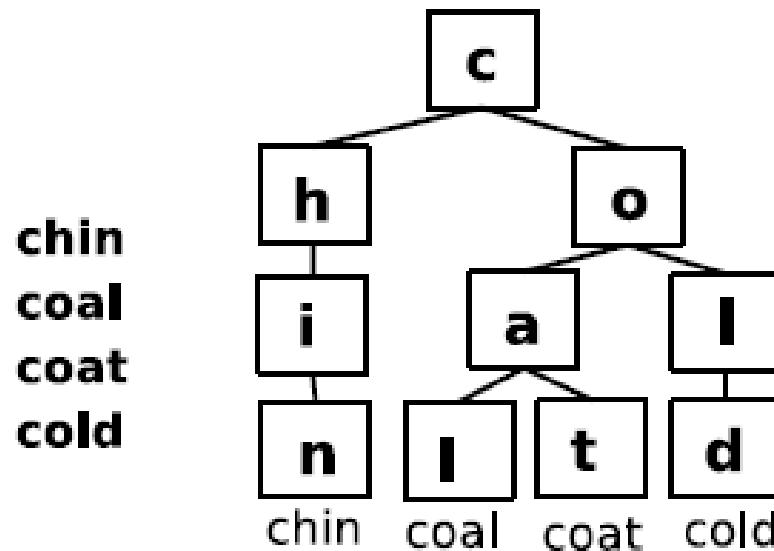
- Find one subgraph at a time (ex: Grochow and Kellis)

A set-centric approach

- **Key insight:** can we do better looking for a given set of subgraphs?
 - All k -subgraphs – even “uninteresting” subgraphs
 - One at a time – no re-usage of computation
 - Can we find what is **common between subgraphs** and use that?
- **Set-centric approach:**
 - Find a custom set of subgraphs
(maybe one, maybe all, maybe something in between)

Inspiration

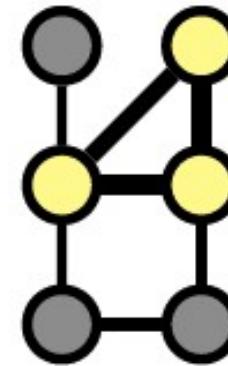
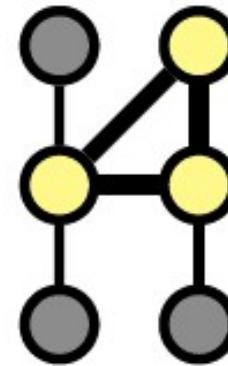
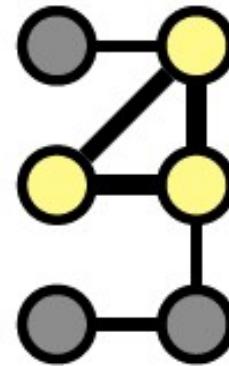
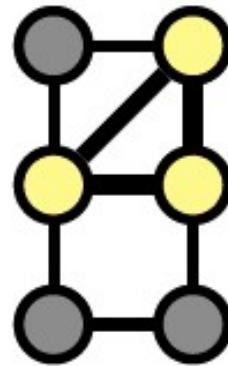
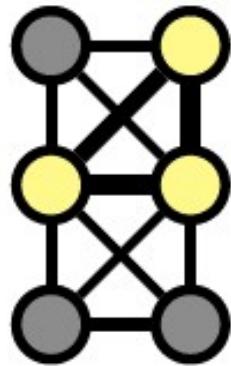
- Sequences and prefix trees



- Can this concept be extended?

Motivation and Concept

- Subgraphs have common substructure



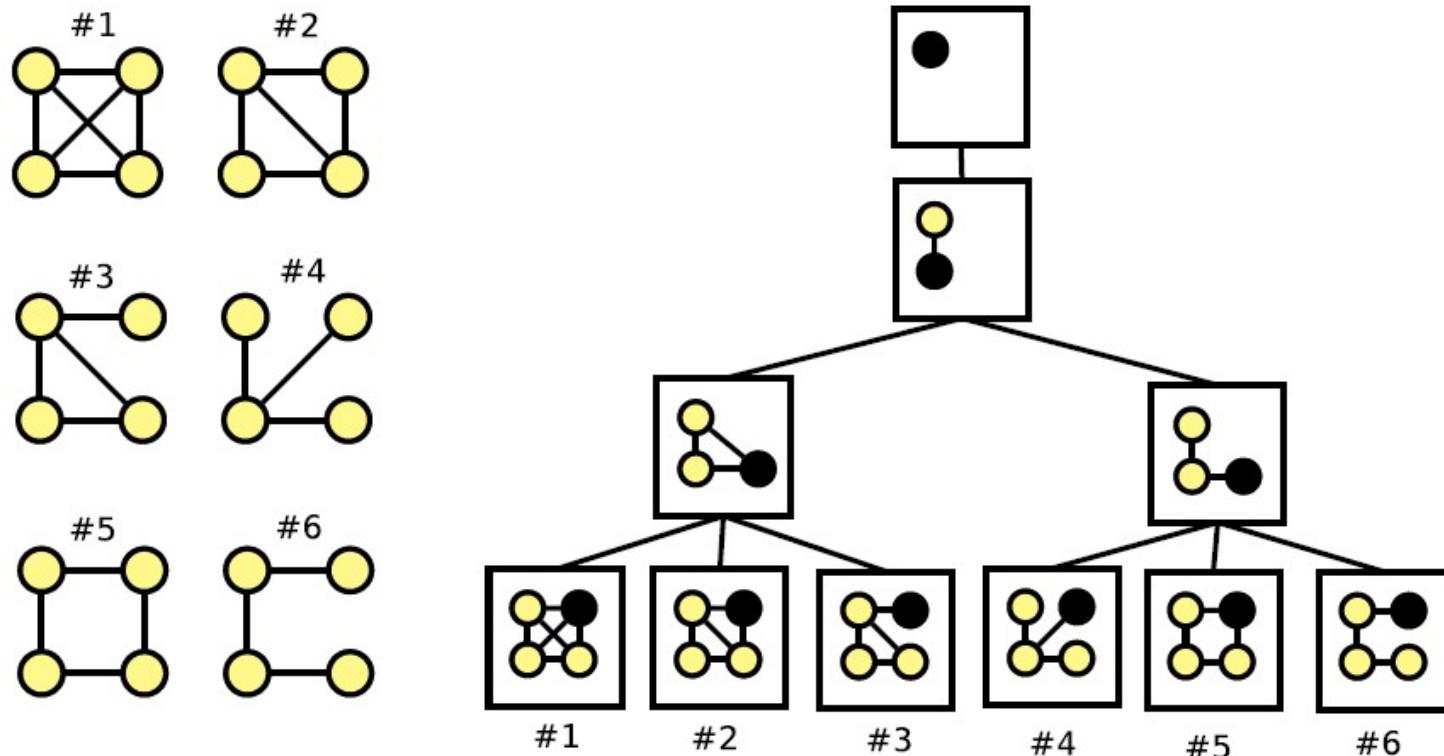
- Create a tree where each tree node corresponds to a single graph vertex

G-Tries

(etymology: Graph Ret TRIE val)

The G-Trie data structure

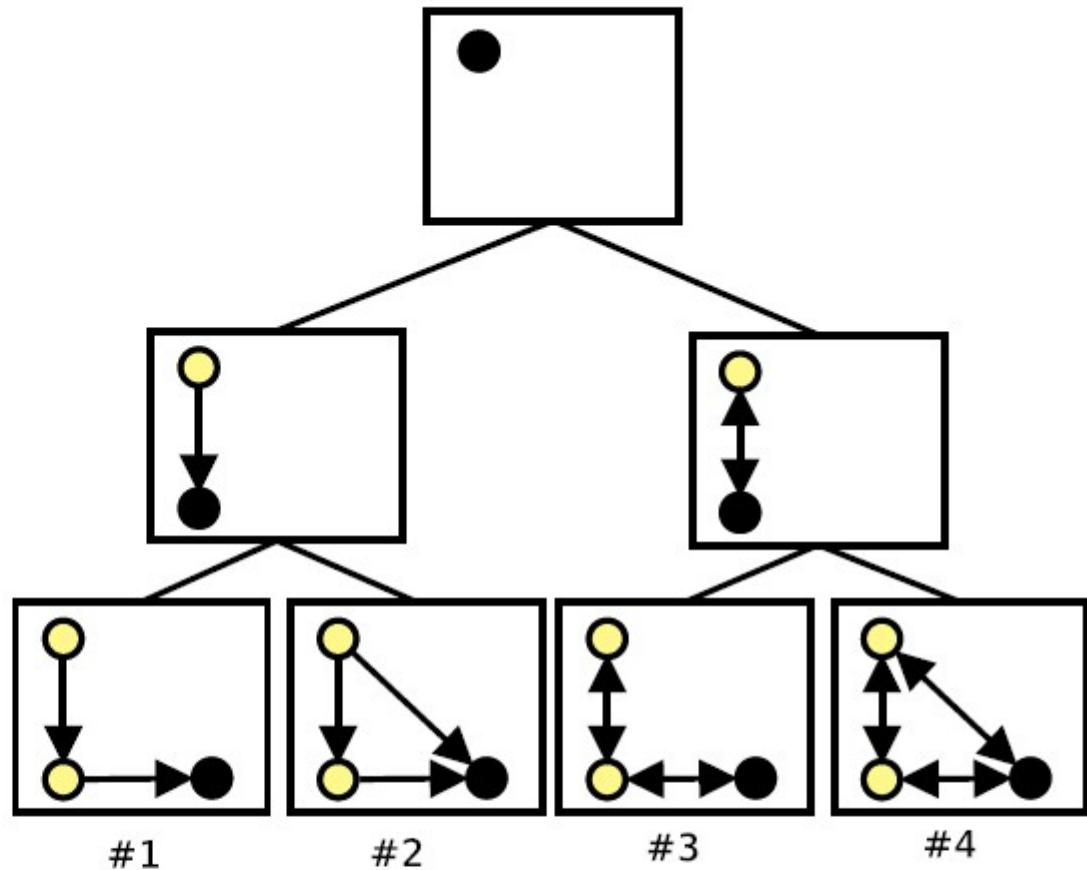
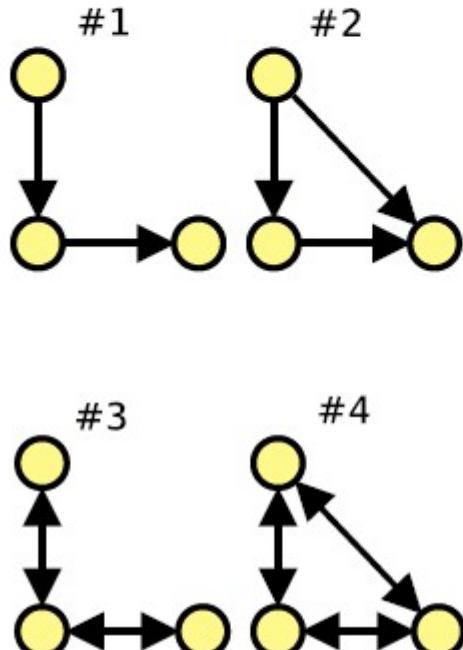
- **G-Tries:** (customized) collections of subgraphs
 - Common substructures are identified
 - Information is “**compressed**”



[Ribeiro & Silva, DMKD, 2014]

The G-Trie data structure

- **G-Tries:** also valid for directed networks



The G-Trie data structure

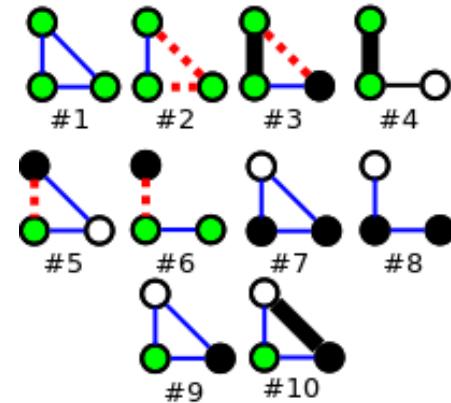
- **G-Tries:** also valid for colored/labeled networks

Subgraph Set

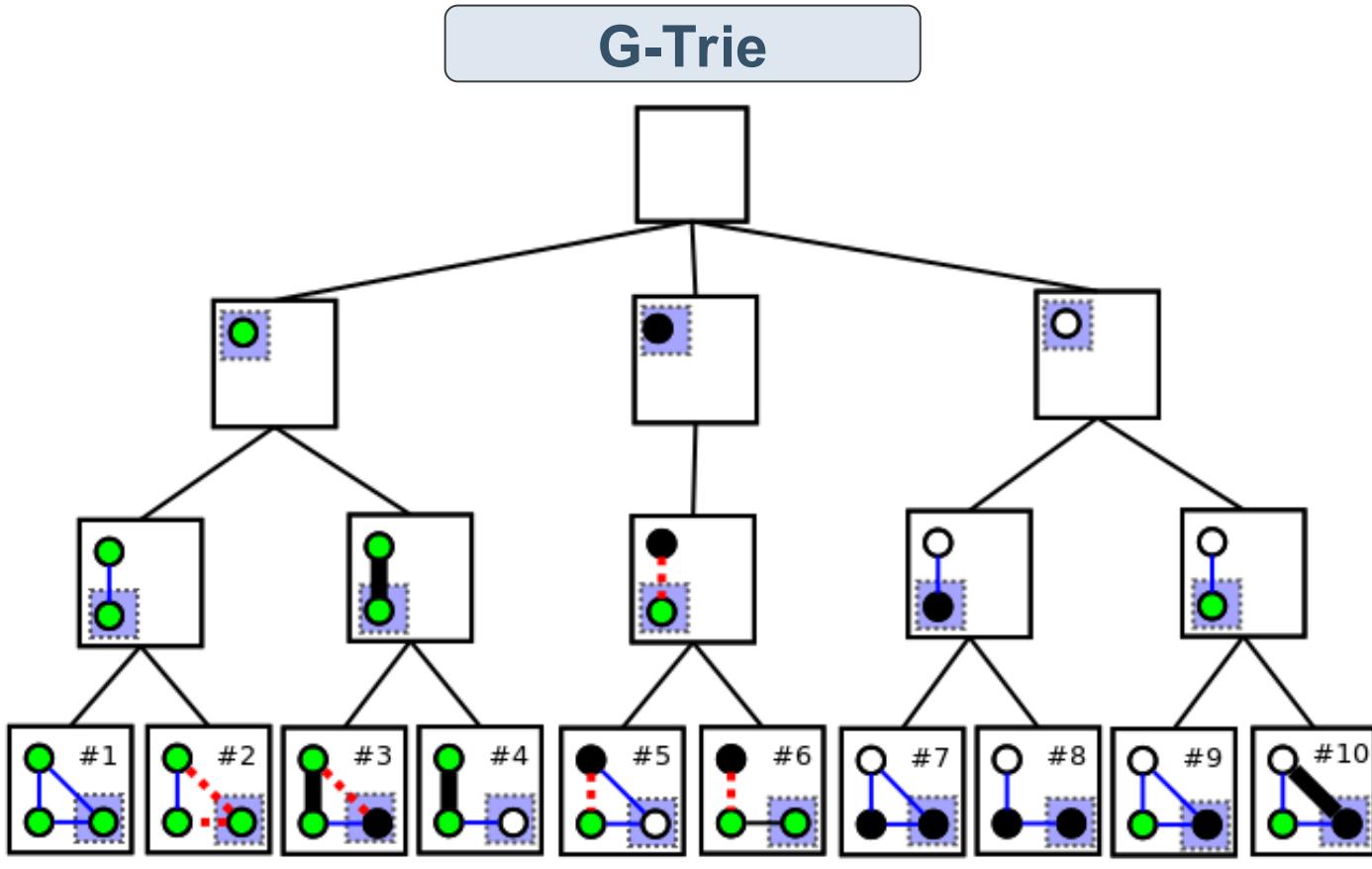
Node types: ● ● ○

Edge types: -----

Subgraphs



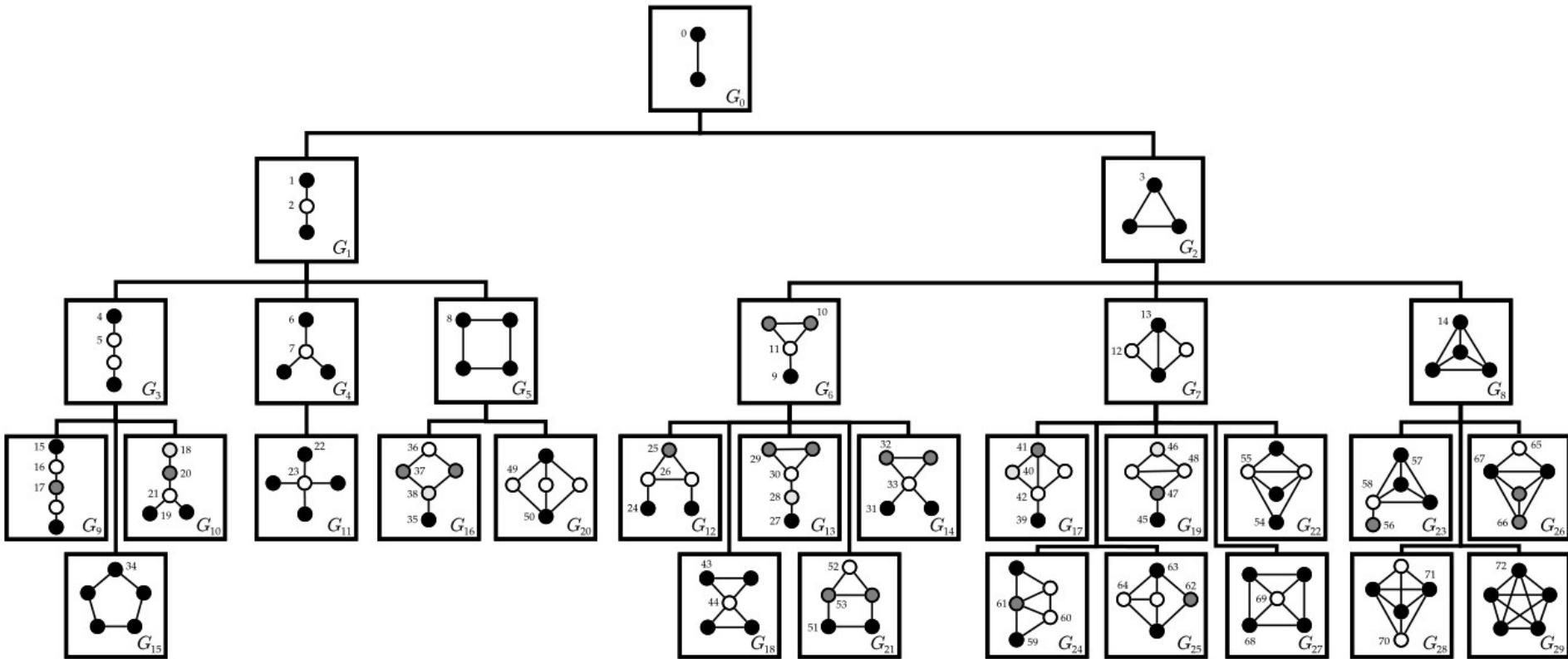
G-Trie



[Ribeiro & Silva, Complenet'2014]

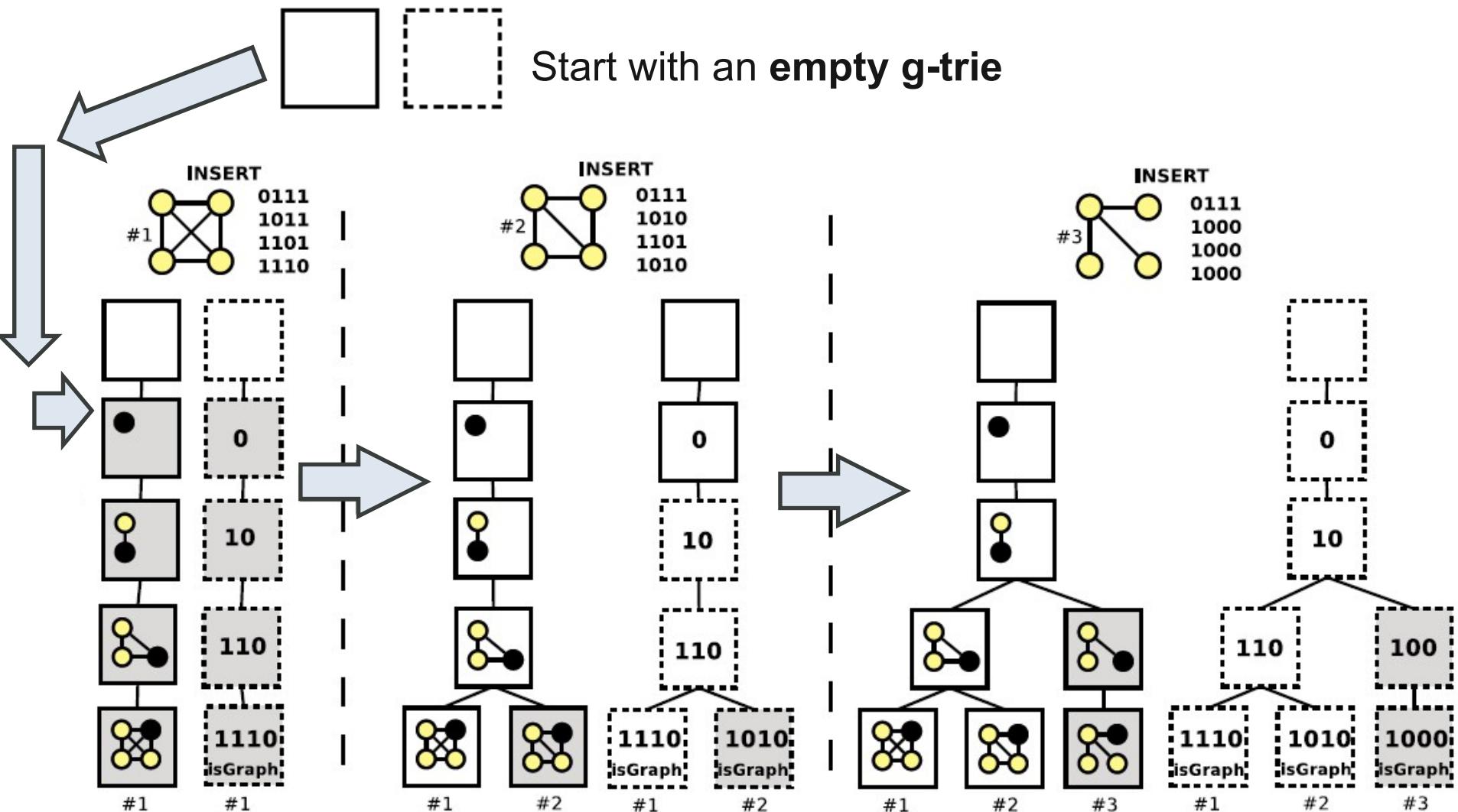
The G-Trie data structure

- **G-Tries:** can also incorporate **orbit** information



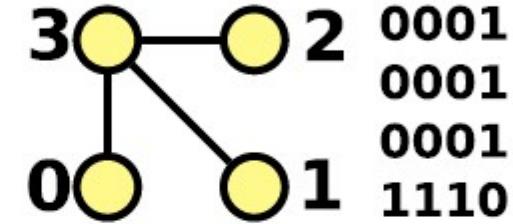
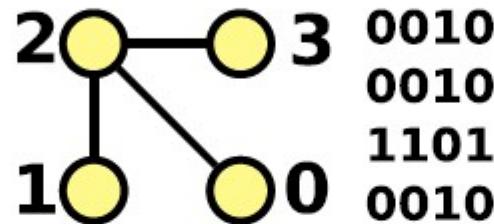
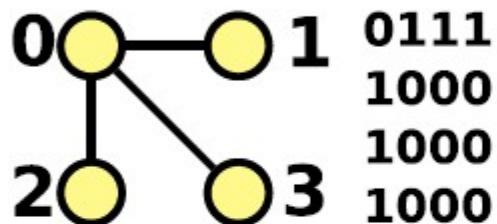
Creating a G-Trie

- Iterative insertion



The Need for a Canonical Form

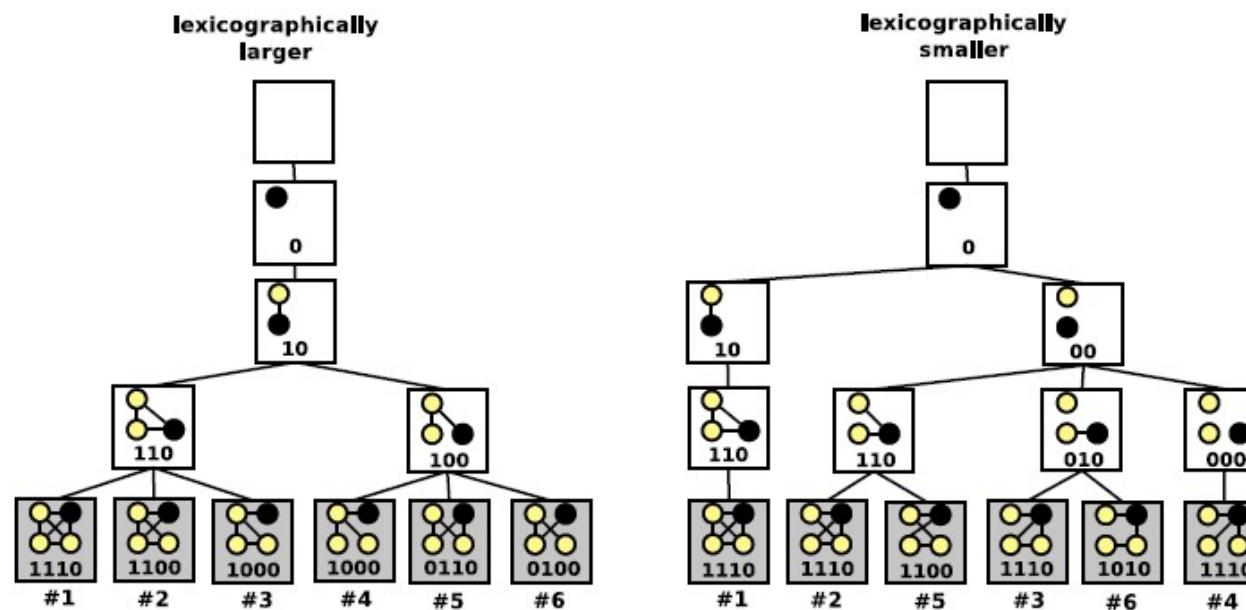
- There are different **node orderings** representing the same subgraph



- **Canonical form** for a getting an unique g-trie
- **Different canon will give origin to different g-tries**

Impact of Canonical Form

Graph	#1	#2	#3	#4	#5	#6
lexicographically larger	0111 1011 1101 1110	0111 1011 1100 1100	0111 1010 1100 1000	0111 1000 1000 1000	0110 1001 1001 0110	0110 1001 1000 0100
lexicographically smaller	0111 1011 1101 1110	0011 0011 1101 1110	0001 0011 0101 1110	0001 0001 0001 1110	0011 0011 1100 1100	0001 0010 0101 1010



Custom Canonical Form

- **Connectivity**

- Path induces connected subgraph

- **Compressibility**

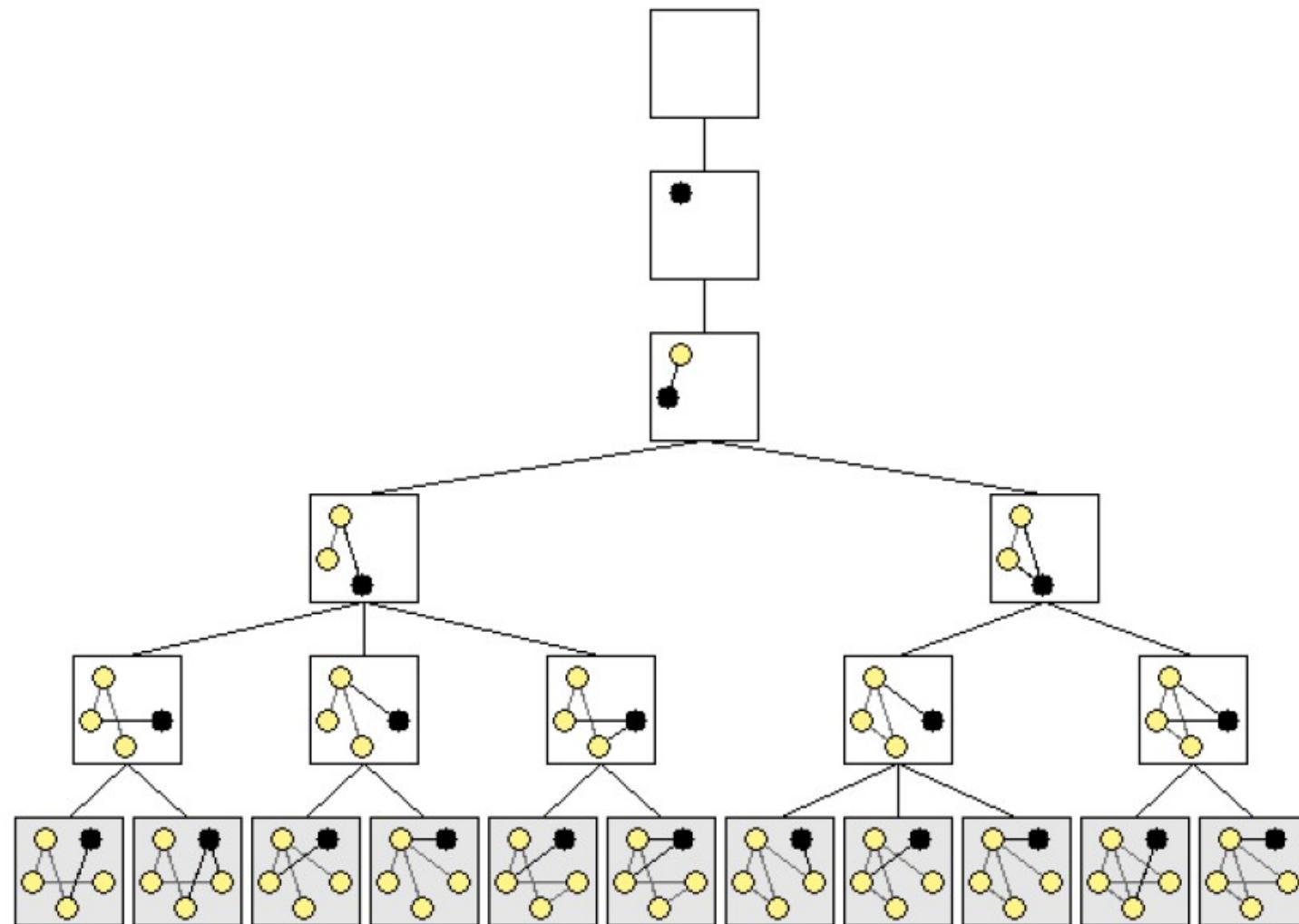
- More common substructure, less g-tries nodes

- **Constraining**

- As many connections as possible to ancestor nodes (limit possible matches)

GTCanon

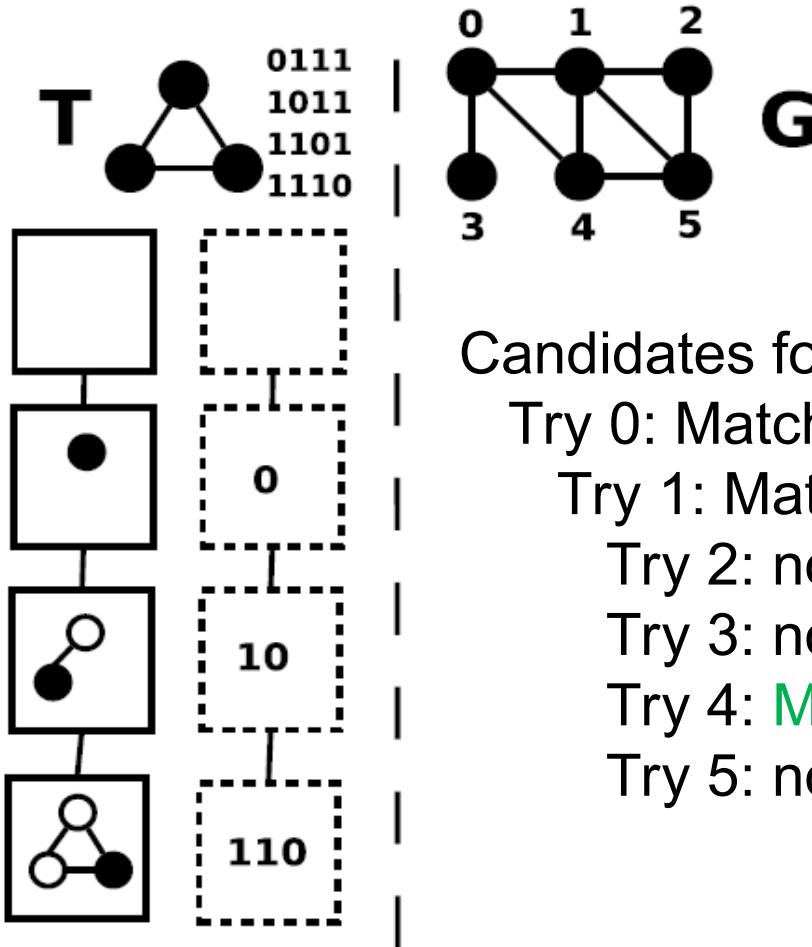
GTCanon Example



Searching with G-Tries

• Backtracking Procedure

- Searching at the same time for several subgraphs



Candidates for node 1: {0, 1, 2, 3, 4, 5}

Try 0: Match = {0}, Neighb. = {1,3,4}

Try 1: Match = {0,1}, Neighb. = {2,3,4,5}

Try 2: no edge from 2 to 0! FAIL

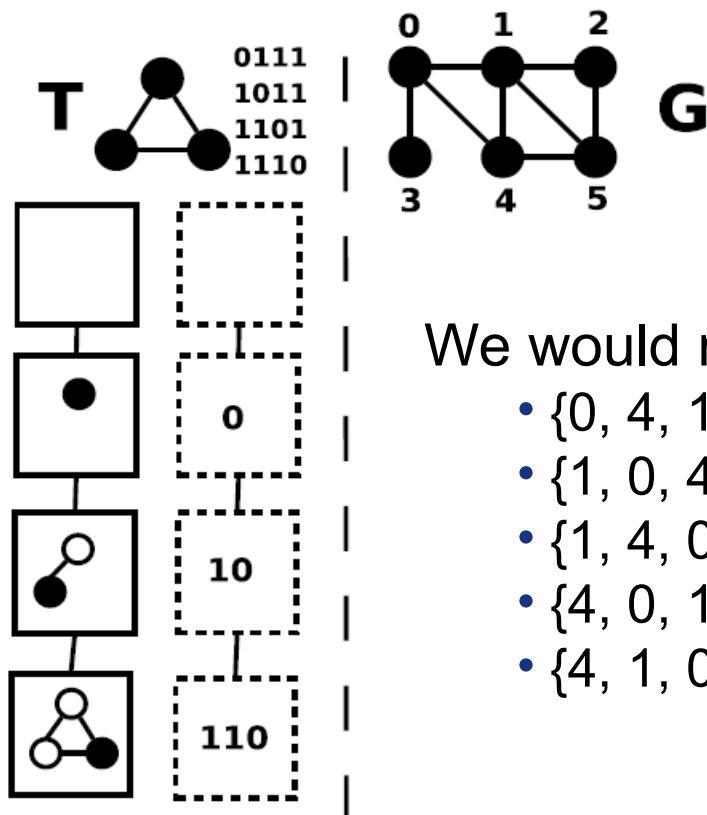
Try 3: no edge from 3 to 1! FAIL

Try 4: Match = {0, 1, 4} FOUND!

Try 5: no edge from 5 to 1! FAIL

Searching with G-Tries

- The same subgraph could be found several times due to automorphisms (symmetries)

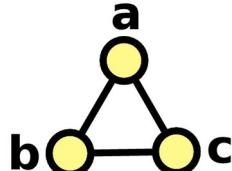


We would not only find $\{0, 1, 4\}$ but also:

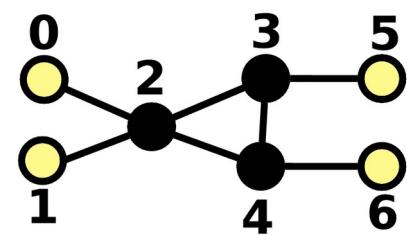
- $\{0, 4, 1\}$
- $\{1, 0, 4\}$
- $\{1, 4, 0\}$
- $\{4, 0, 1\}$
- $\{4, 1, 0\}$

Symmetry Breaking Conditions

- **Conditions on node labels**



Symmetry Breaking Conditions: $\{a < b, b < c\}$



Possible Matches of $\{a,b,c\}$ in the graph of size 7:

$\{2,3,4\}$ - OK!

$\{2,4,3\}$ - No match ($b > c$)

$\{3,2,4\}$ - No match ($a > b$)

$\{3,4,2\}$ - No match ($b > c$)

$\{4,2,3\}$ - No match ($a > b$)

$\{4,3,2\}$ - No match ($a > b, b > c$)

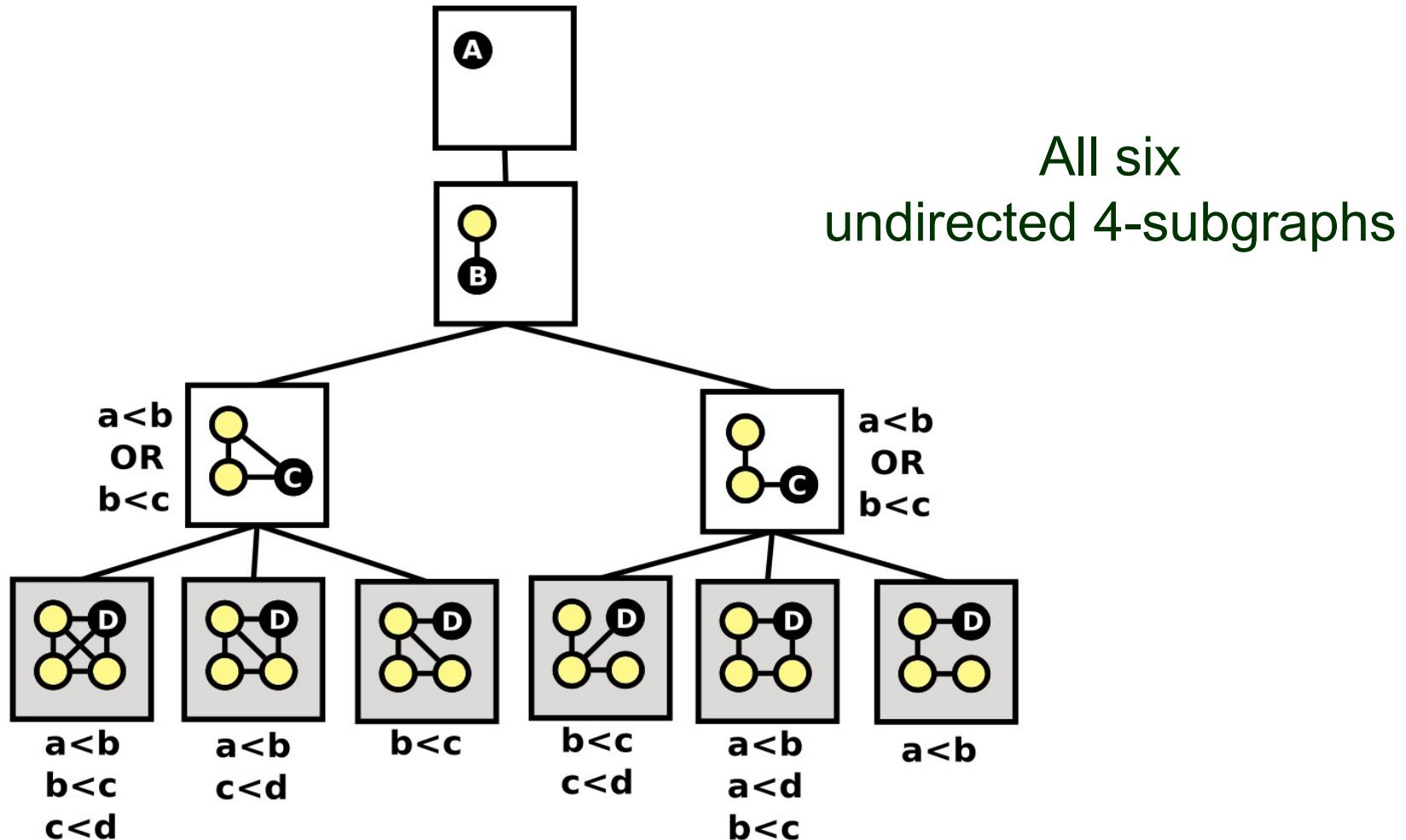
- **Augment g-trie with these conditions**

- Match only when conditions of at least one descendant are respected

- **Filter conditions to ensure minimum work**

- Ex: transitive property ($a < b, a < c, b < c$ leads to $a < b, b < c$); assured descendants, only store relevant to node, etc

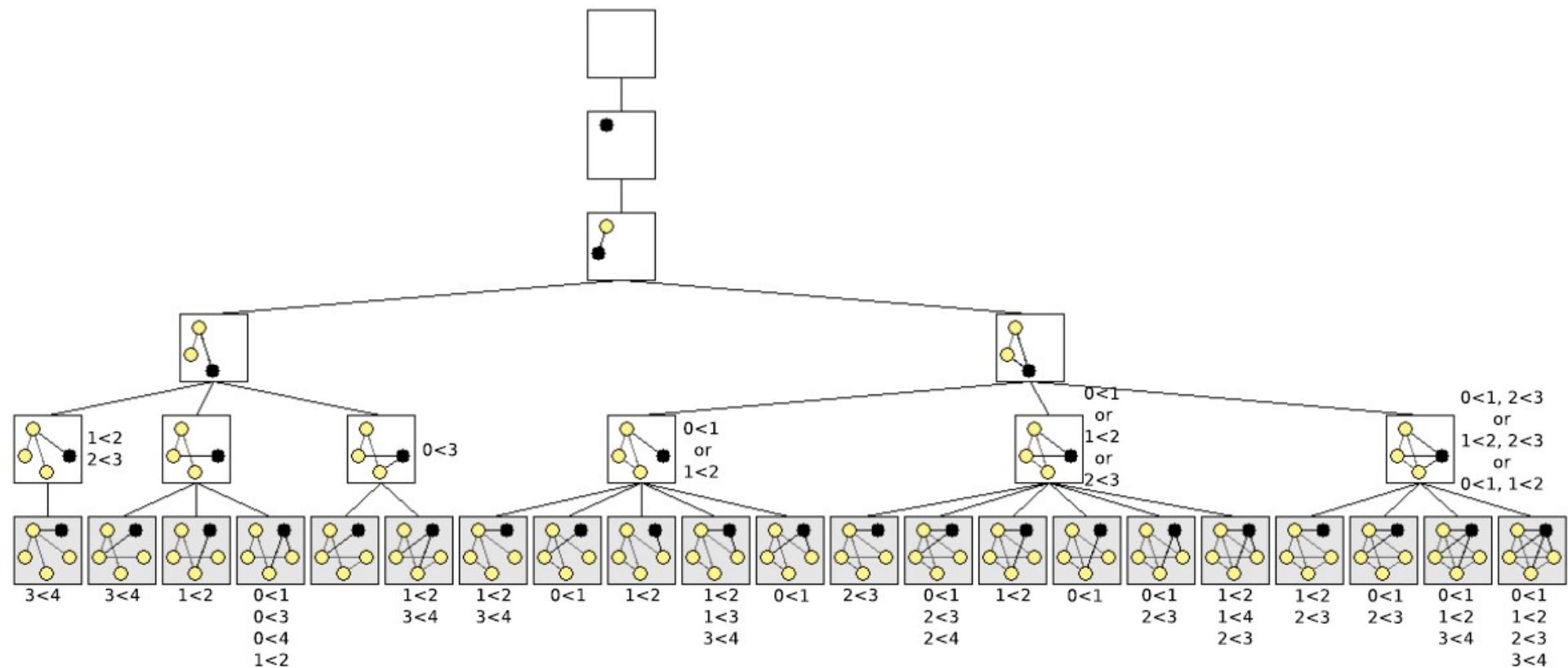
Complete G-Trie Example



All six
undirected 4-subgraphs

Complete G-Trie Example

All 21 undirected 5-subgraphs



Sequential version: some results

- **Comparison with main competing algorithms**
 - ESU & Kavosh (network-centric)
 - Grochow and Kellys (subgraph-centric)
- **Implemented in common framework**
 - Implementation at least as efficient as original
 - C++ as the programming language
 - Efficient graph primitives
 - More “fair” comparison

Sequential version: some results

- Set of 12 representative networks

Network	Group	Directed	V(G)	E(G)	Nr. Neighbours	
					Average	Max
dolphins	social	no	62	159	5.1	12
circuit	physical	no	252	399	3.2	14
neural	biological	yes	297	2,345	14.5	134
metabolic	biological	yes	453	2,025	8.9	237
links	social	yes	1,490	19,022	22.4	351
coauthors	social	no	1,589	2,742	3.5	34
ppi	biological	no	2,361	6,646	5.6	64
odlis	semantic	yes	2,909	18,241	11.3	592
power	physical	no	4,941	6,594	2.7	19
company	social	yes	8,497	6,724	1.6	552
foldoc	Semantic	yes	13,356	120,238	13.7	728
internet	Physical	no	22,963	48,436	4.2	2,390

Sequential version: some results

- On both directed and undirected graphs we were from 1 to 2 orders of magnitude faster than existing state of the art at that time
 - From 10x to 200x

Example results for **full census of size k**
(speedup on a set of undirected networks)

Network	k	ESU	Kavosh	Grochow
dolphins	8	28.9	26.9	39.5
circuit	9	53.2	52.0	39.4
coauthors	6	64.4	66.3	39.7
ppi	5	61.8	62.1	25.6
power	7	38.2	38.0	285.9
internet	4	46.9	45.5	14.7

Sequential version: some results

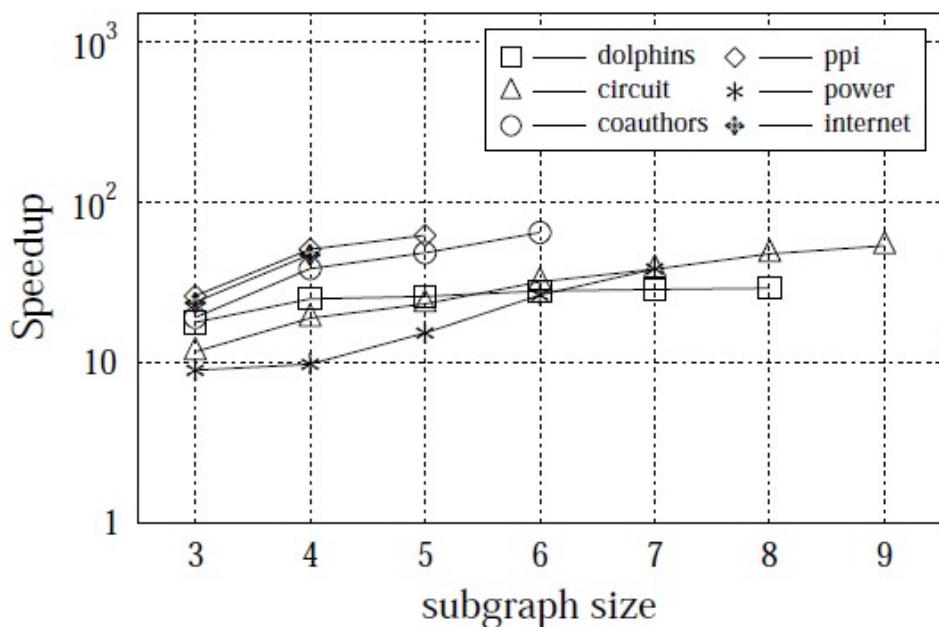
- On both directed and undirected graphs we were from 1 to 2 orders of magnitude faster than existing state of the art at that time
 - From 10x to 200x

Example results for **full census of size k**
(speedup on a set of directed networks)

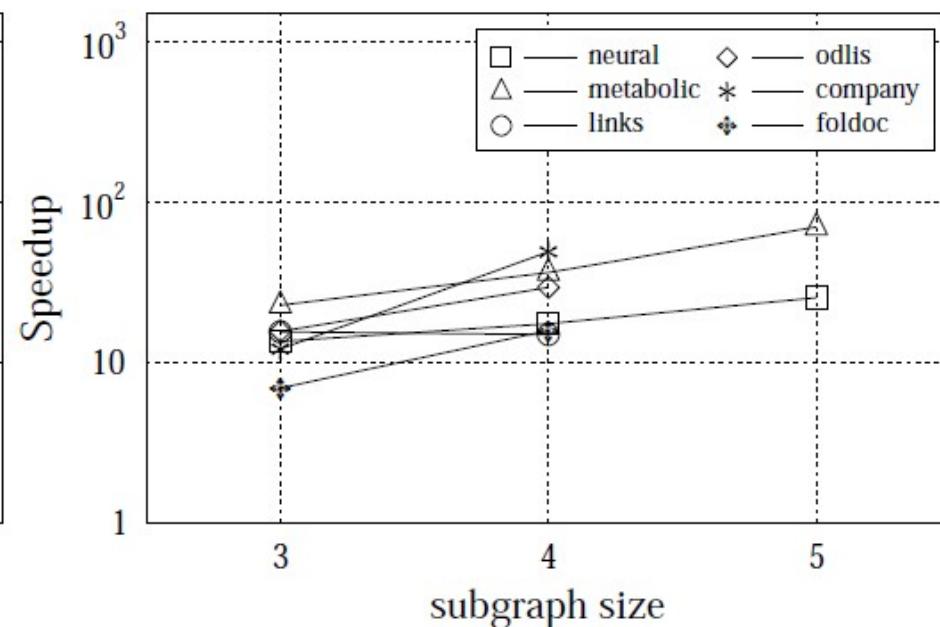
Network	K	ESU	Kavosh	Grochow
neural	5	25.3	25.5	28.8
metabolic	5	69.9	68.9	15.4
links	4	14.9	15.2	13.2
odlis	4	29.3	29.7	22.6
company	4	48.9	50.1	25.3
foldoc	4	15.8	16.0	50.5

Sequential version: some results

- On both directed and undirected graphs we were from 1 to 2 orders of magnitude faster than existing state of the art at that time
 - From 10x to 200x



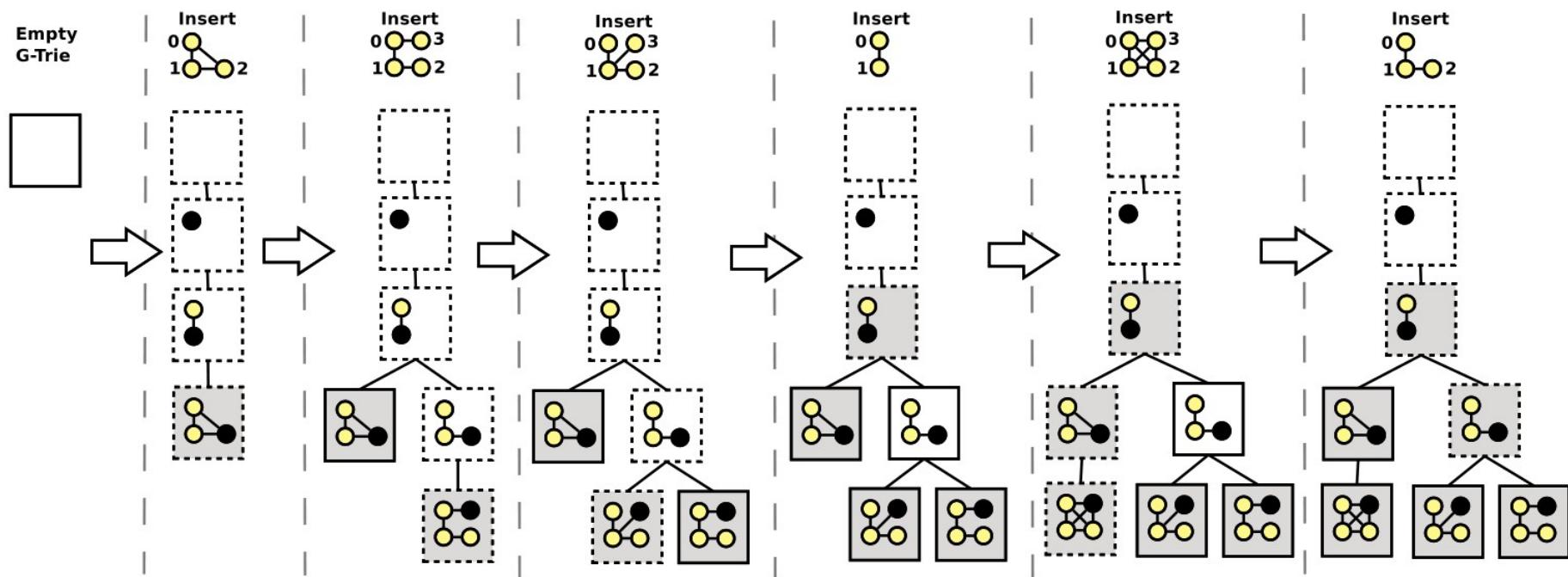
(a) vs ESU on undirected networks



(b) vs ESU on directed networks

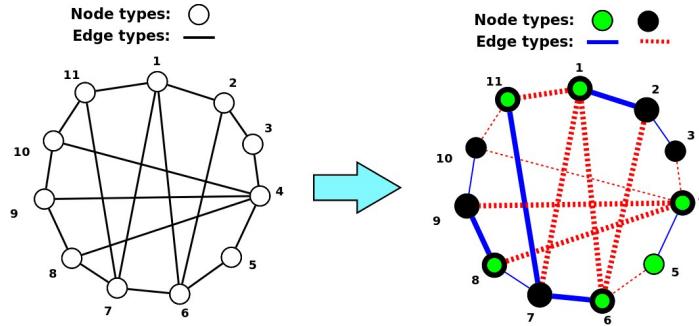
Sequential version: some results

- Speedup also when looking for **different sets of subgraphs** (other than full census of size k)
 - Better speedup as more subgraphs are being searched at the same time (**set-centric**)



Sequential version: some results

- Speedup also when using colored networks

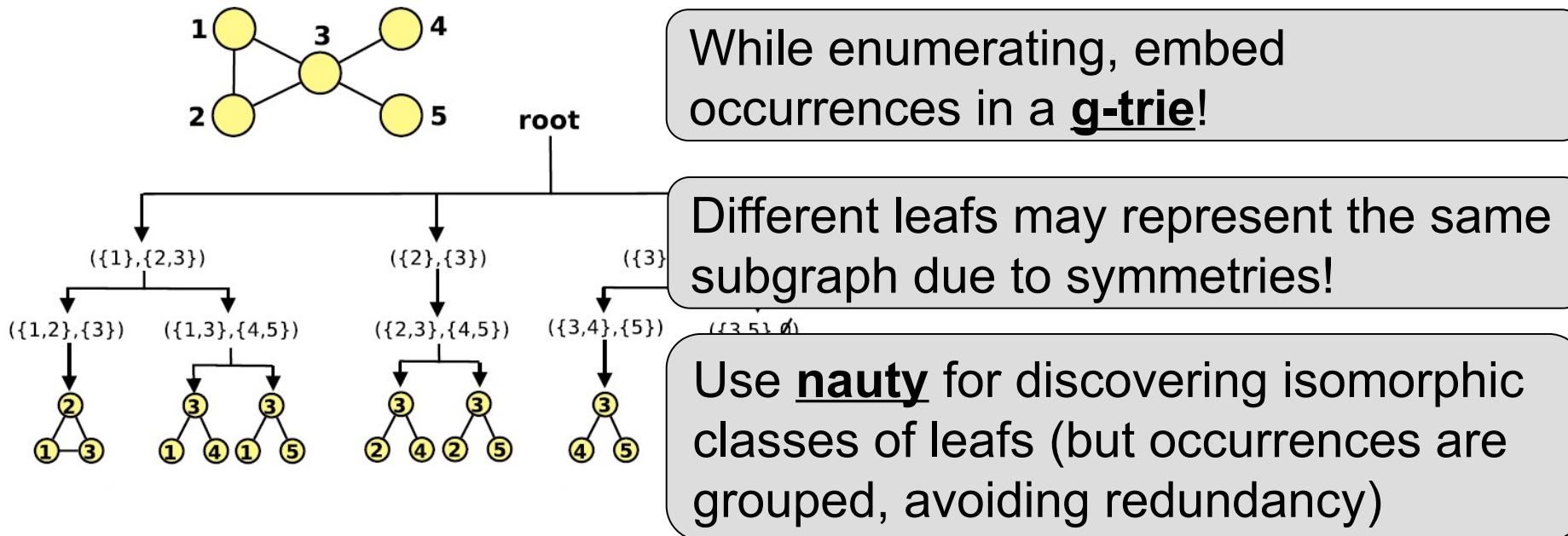


network	k	Execution Time (seconds)						Speedup G-Tries vs ESU		
		ESU (via Fanmod)			G-Tries					
		Original	Avg.	Random	Total	Original	Avg.	Random	Total	
blogs	3	2.1	2.1	209.06	209.06	0.73	0.29	29.73	29.73	7.0x
	4	232.10	263.45	26,577.10	26,577.10	53.04	15.10	1,563.04	1,563.04	17.0x
dblp	3	0.50	0.25	25.50	25.50	0.15	0.02	2.15	2.15	11.9x
	4	8.11	11.80	1,188.11	1,188.11	1.90	0.17	18.90	18.90	62.9x
	5	276.03	479.57	48,233.03	48,233.03	70.02	5.50	620.02	620.02	77.8x
flights	3	1.59	1.63	164.59	164.59	0.48	0.05	5.48	5.48	30.0x
	4	139.36	187.00	18,839.36	18,839.36	35.01	4.23	458.01	458.01	41.1x
elections	3	23.02	33.55	3,378.02	3,378.02	7.51	1.70	177.51	177.51	19.0x
	4	6,987.34	7,434.25	750,412.02	750,412.02	800.86	256.68	26,468.85	26,468.85	28.4x

Dynamic G-Tries

- Speedup also when adapting to network-centric methodology

- Use as base any enumeration method (e.g. ESU)

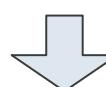


FaSE – Fast Subgraph Enumeration

G-Trie Dynamically Built

[Paredes & Ribeiro, ASONAM' 2013]

Graph Representations

- **Core graph primitive is edge verification**
 - Adjacency Matrix (AdjMat) gives that in $O(1)$
 - Used when $O(n^2)$ fits in memory
- **For larger sparse graphs we use an hybrid representation:**
 - Combine linear search + hash tables + trie
 - Low-level optimizations (cache, bitwise ops, ...)
- [Paredes & Ribeiro, NetSciX'2016]
- **Overhead with AdjMat is small !**
 - From 4x more with binary search
- 
- **Less than 1.5x on average with hybrid approach**

Iterative updates

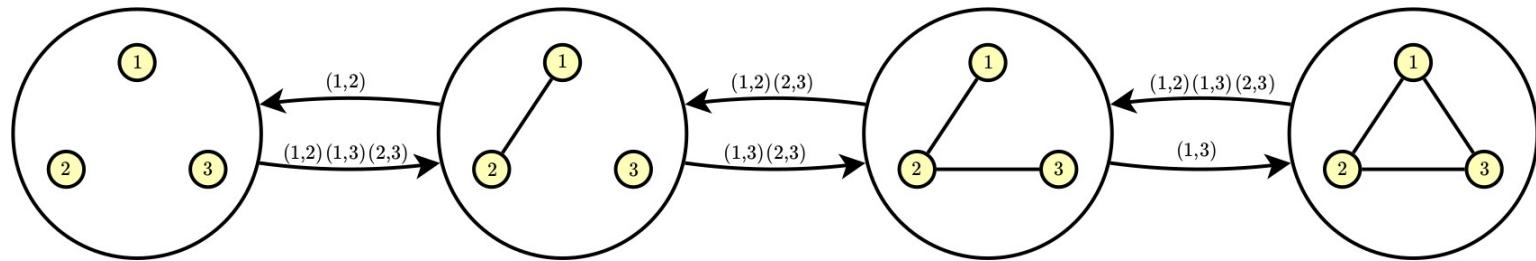
- **Update subgraph counts after edge deletion or removal**

- Take into account only the subgraphs that touch(ed) that particular edge

[Silva, Paredes & Ribeiro, Complenet'2017]

- **Add the capability of following the isomorphic type of a set of nodes**

- Edge updates change the type of subgraph



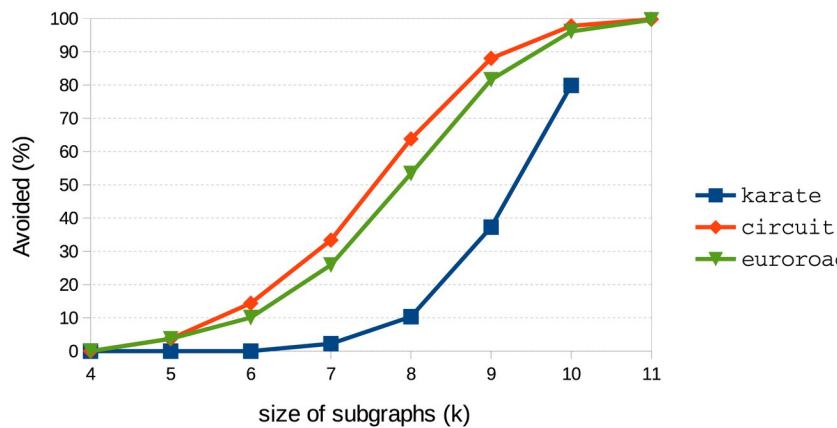
Automaton to keep subgraph type as “state”

[Paredes & Ribeiro, Complenet'2018]

Improve motif discovery

- **Iterative deepening of subgraph size**

- Start with smaller sizes and keep incrementing
- Discard supergraphs that contain *non-interesting* subgraphs (ex: frequency = 0)
- Generate only supergraphs of *interesting* subgraphs



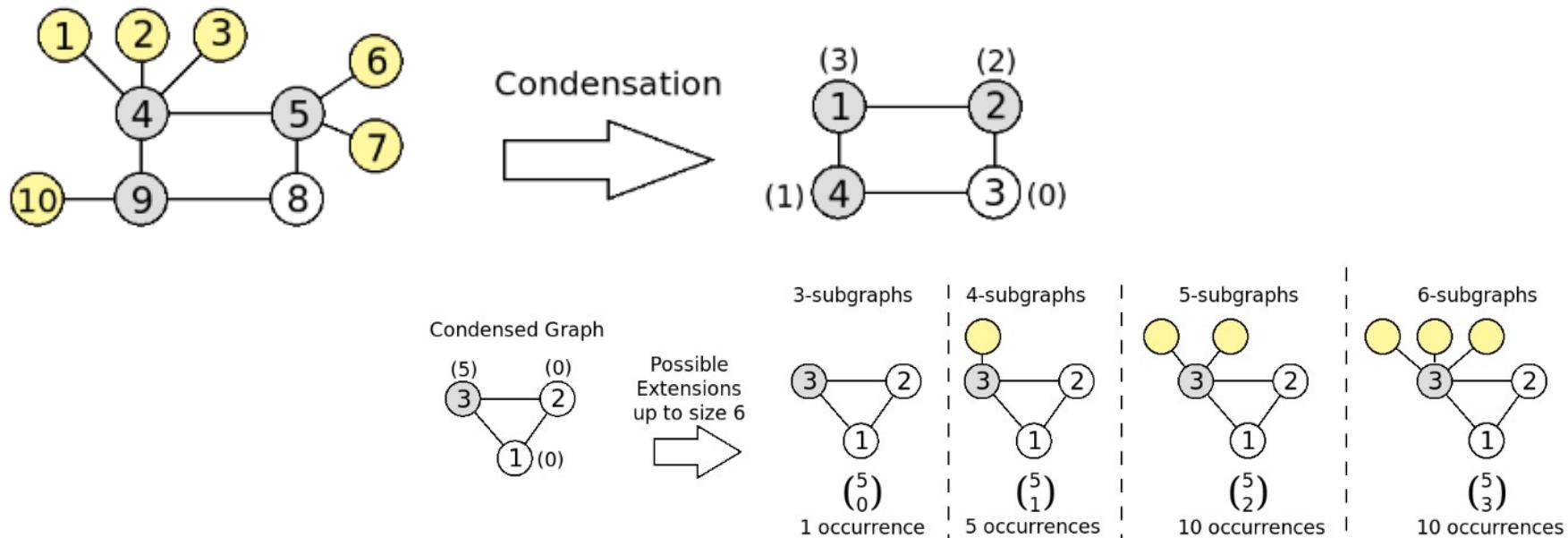
Improve candidate subgraph generation

[Grácio & Ribeiro, Complenet'2019]

Improve motif discovery

- **Combinatorial optimizations**

- Lossless compression of original graph
- Count on reduced graph; extrapolate results

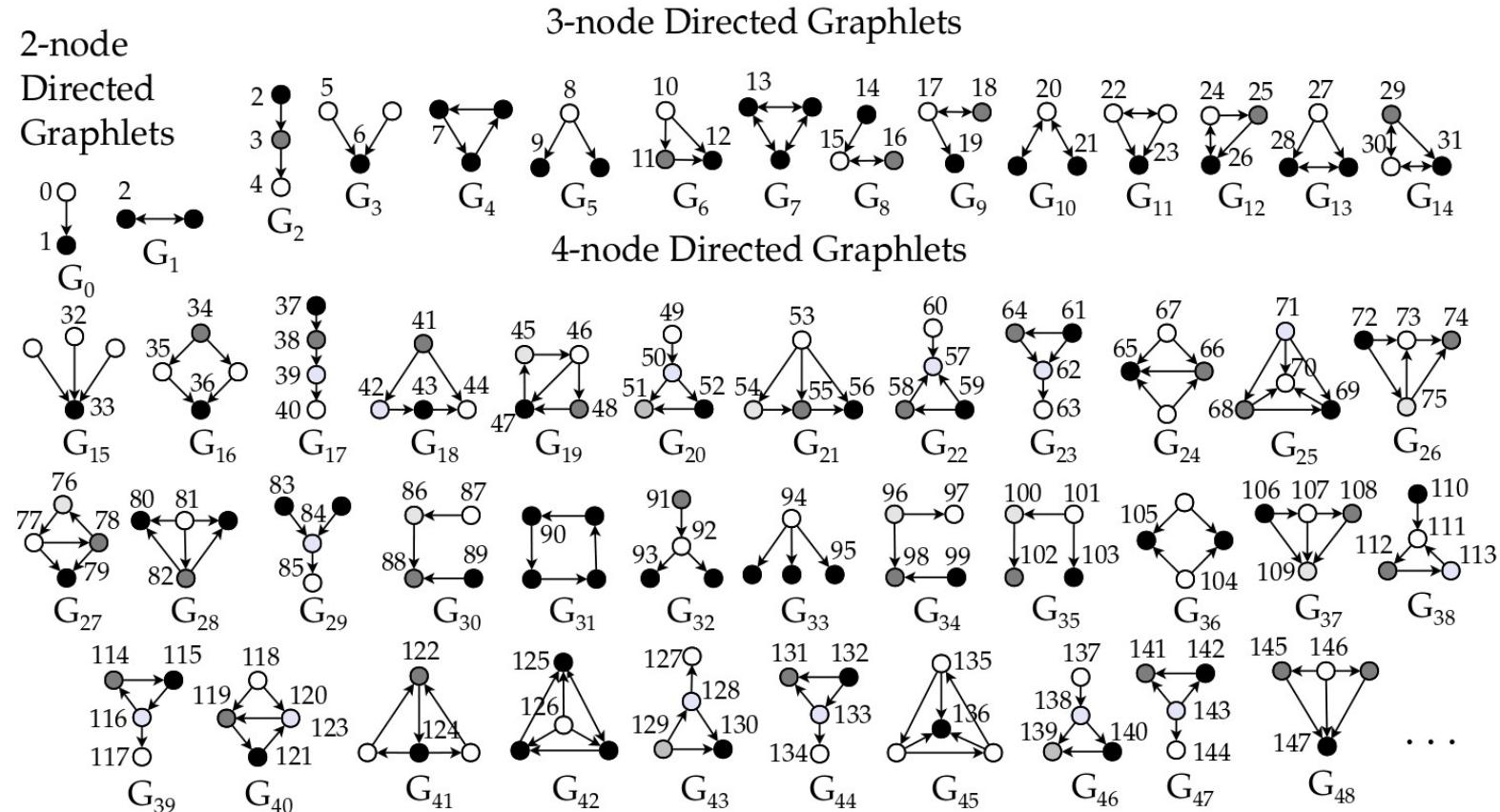


Account for multiple occurrences once

[Martins & Ribeiro, Complenet'2020]

Extending existing metrics

- Extending the applicability of graphlets to directed networks

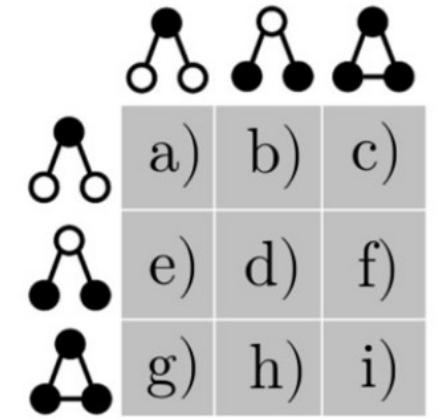
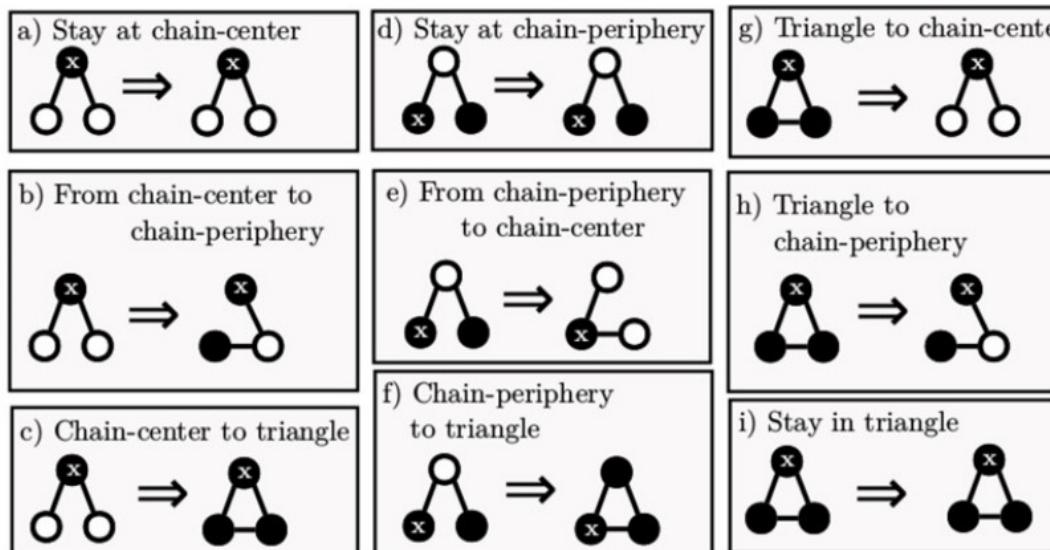
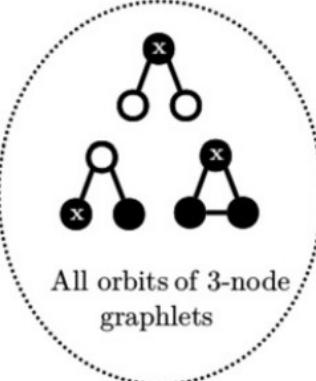


[Aparício, Ribeiro & Silva, TCBB, 2017]

Temporal networks

- Study evolution of subgraphs

Possible Graphlet-Orbit Transitions of node

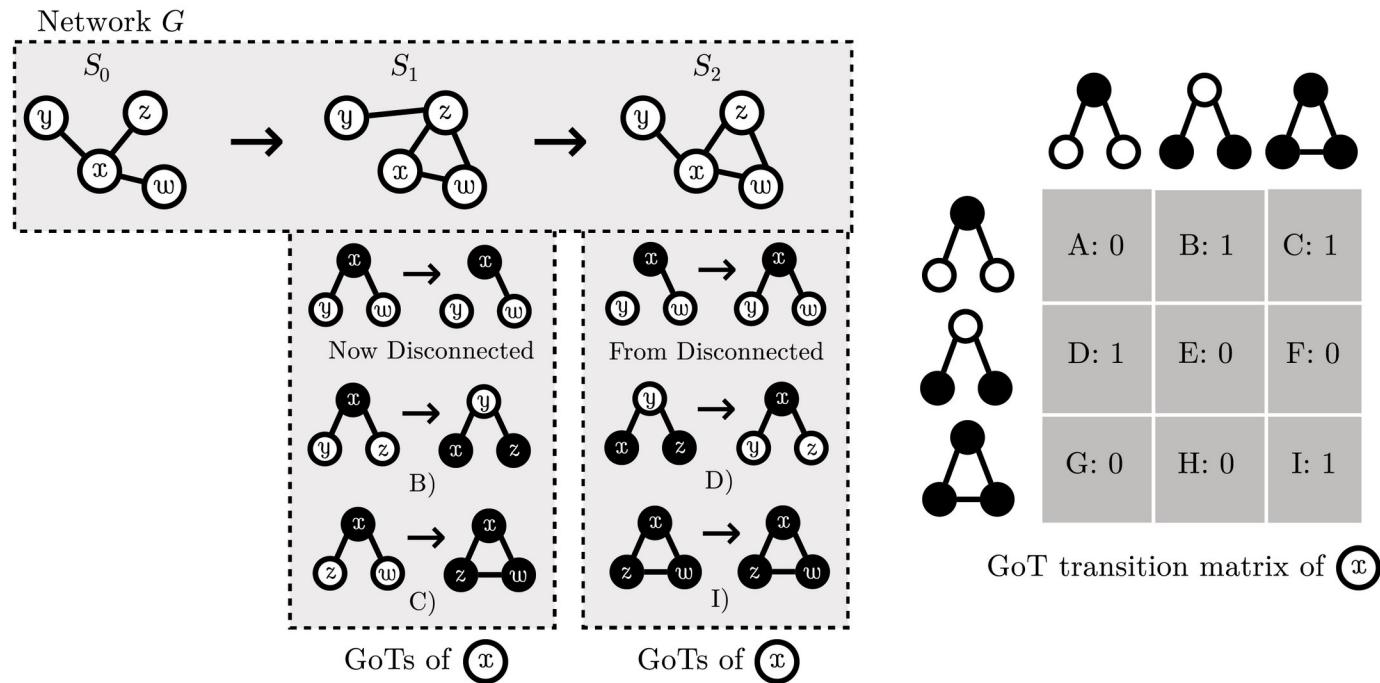


Graphlet-Orbit Transitions (**GoT**): fingerprints for temporal network comparison

[Aparício, Ribeiro & Silva, PloS One, 2018]

Temporal networks

- Study evolution of subgraphs

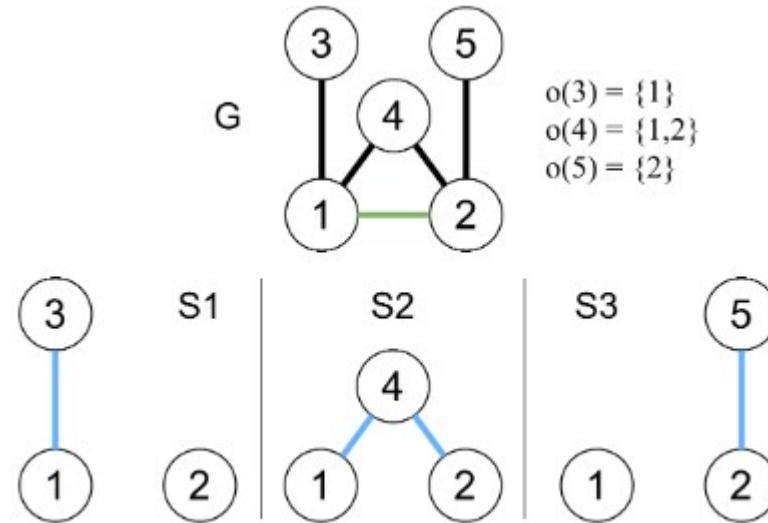
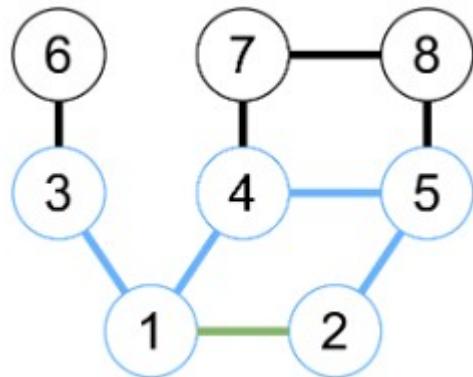


Graphlet-Orbit Transitions (**GoT**): fingerprints for temporal network comparison

[Aparício, Ribeiro & Silva, PloS One, 2018]

Temporal networks

- Counting in streaming networks

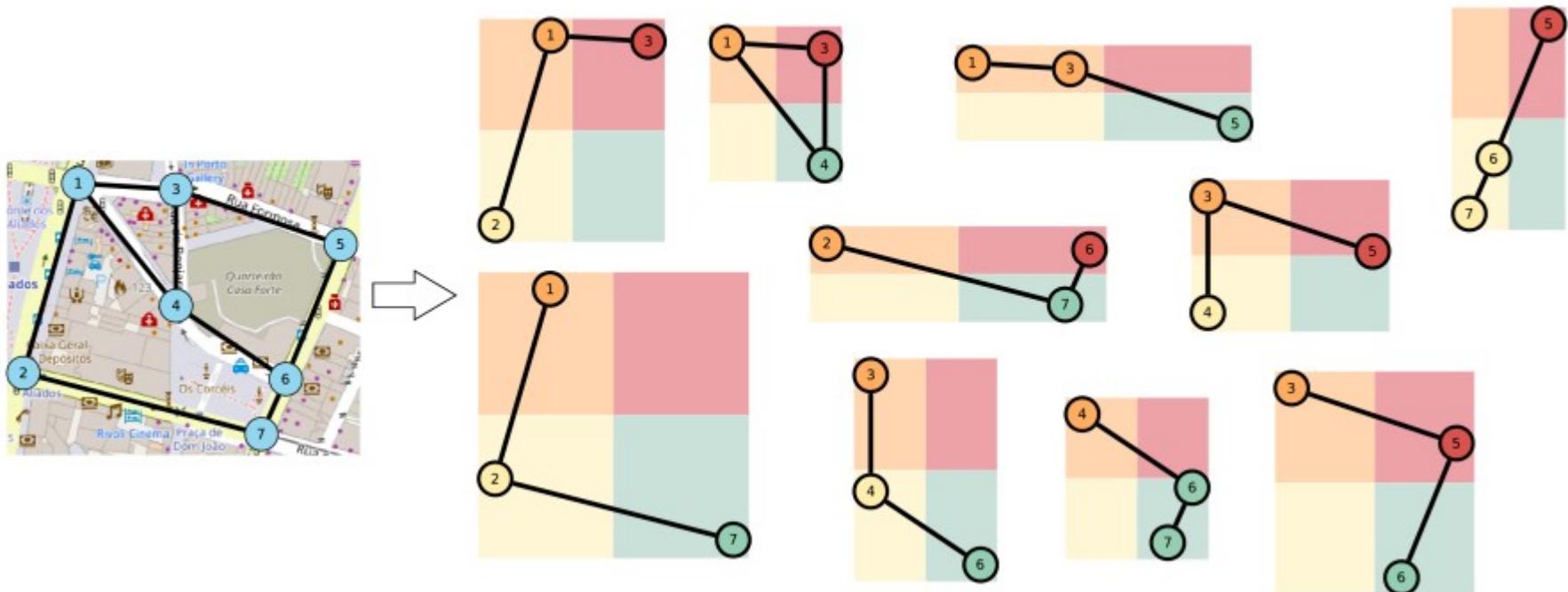


StreamFaSE: An online algorithm for
subgraph counting in dynamic networks

[Branquinho, Grácio and Ribeiro, CNA, 2020]

Spatial Networks

- **Networks with spatial features**

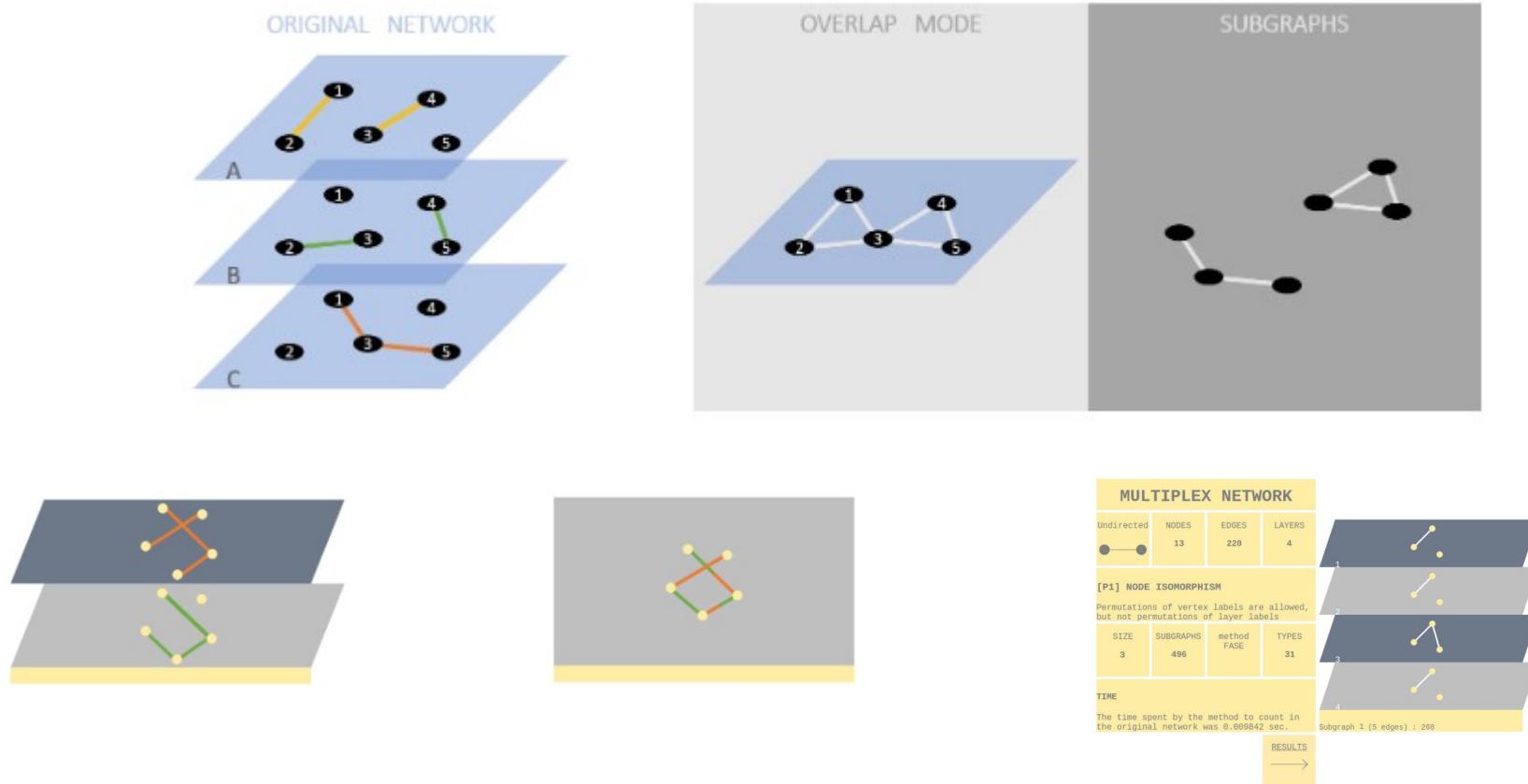


Towards the Concept of Spatial Network Motifs

[Ferreira, Barbosa and Ribeiro, CNA, 2022]

Multilayer Networks

• Motifs in networks with multiple layers



Journal submission being prepared

[Meira & Ribeiro, *in preparation*]

Higher dimensions

• Hypergraphs

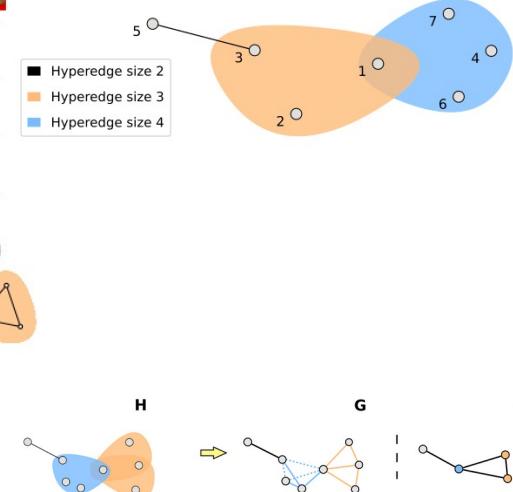
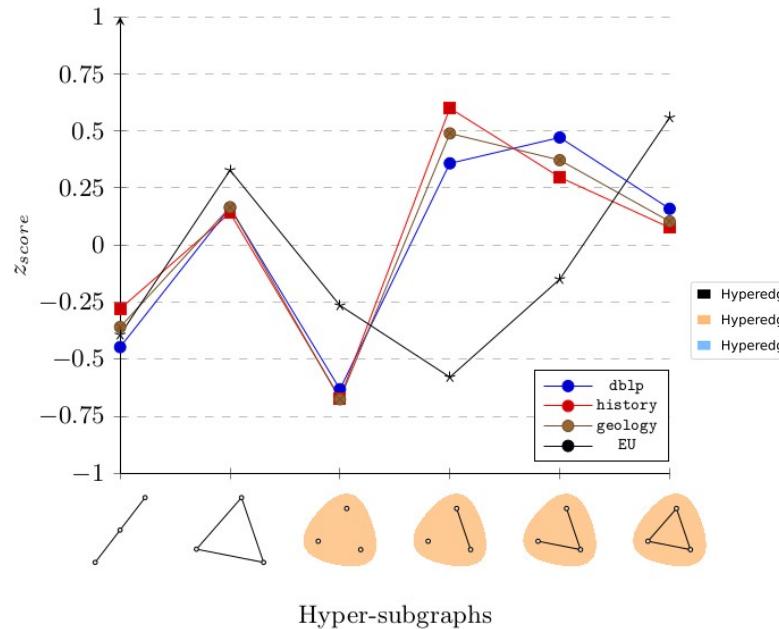
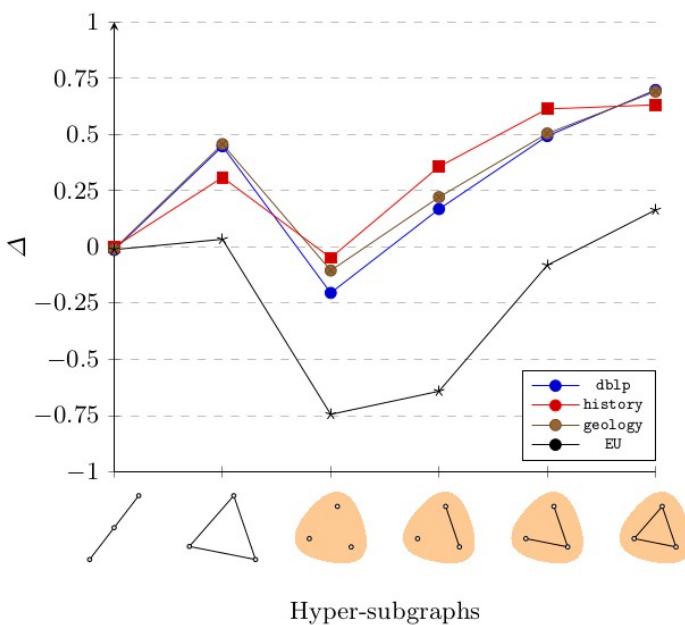


Fig. 2: Hypergraph conversion to projection graphs.

Computing Motifs in Hypergraphs

[Nóbrega and Ribeiro, Complenet'2024]

4) SAMPLING APPROACH

Approximating results

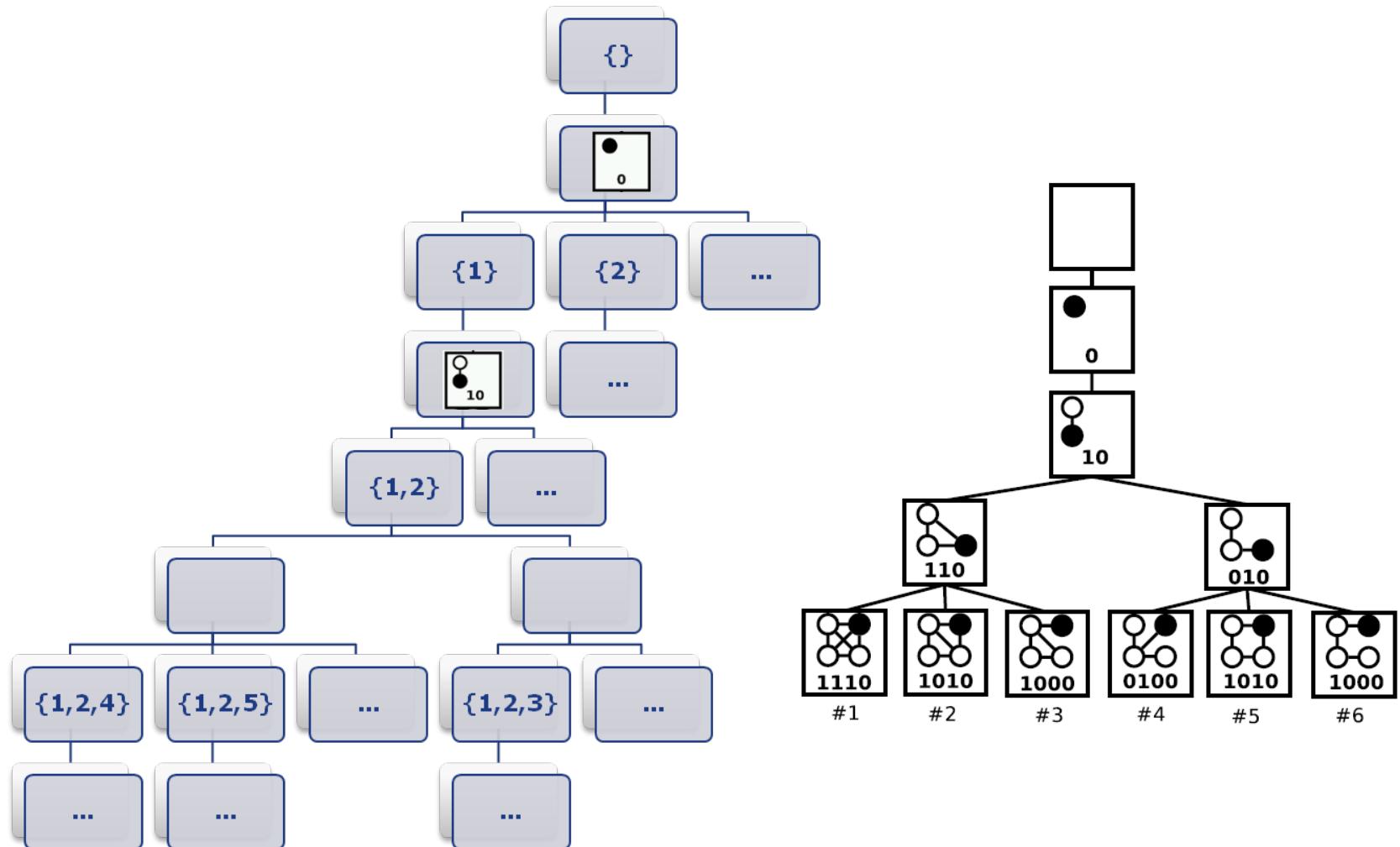
- **Sample subgraph occurrences**

- Compute approximate results
- Trade **accuracy** for **speed**



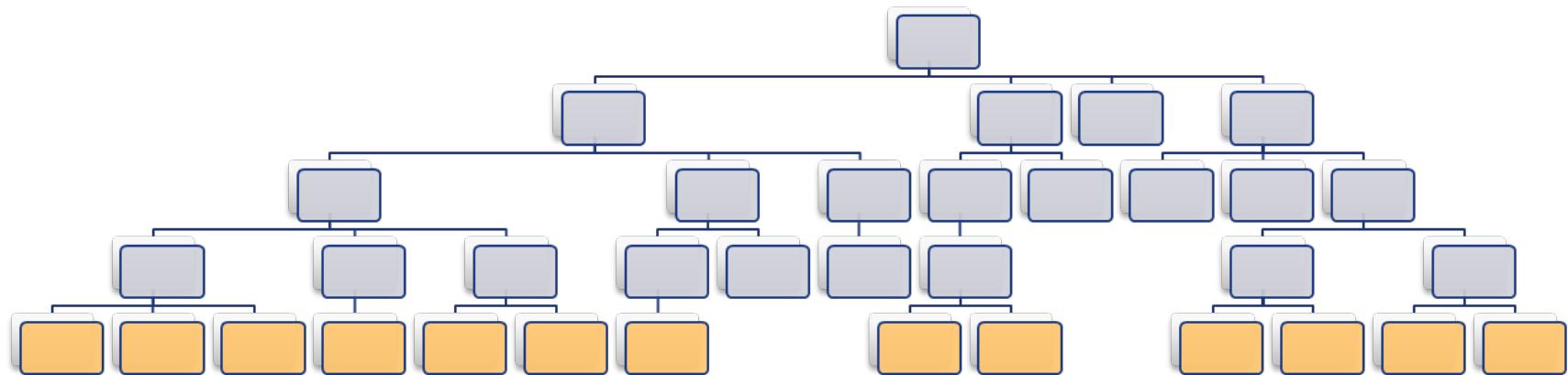
Sampling approach

- Backtracking procedure produces search tree



Sampling approach

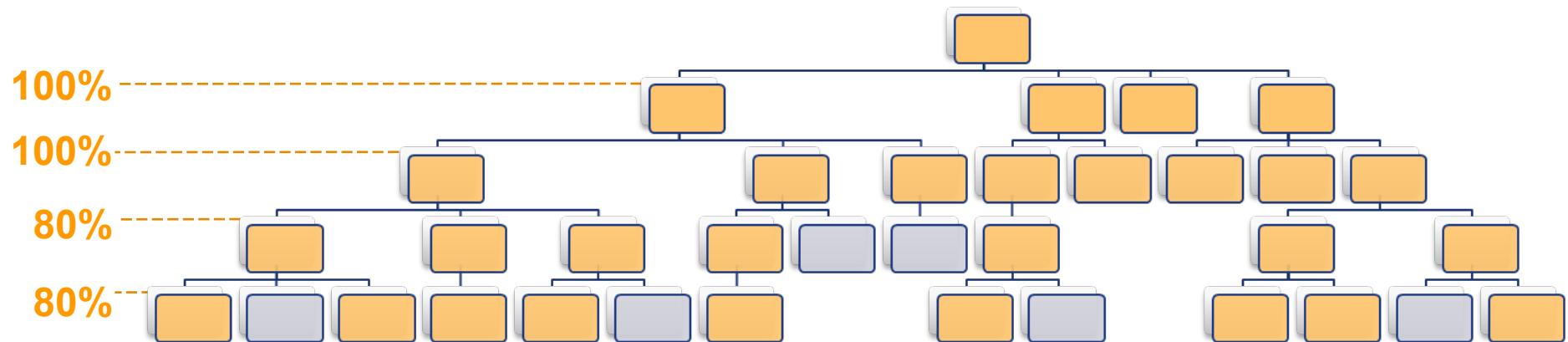
- **Original:** unbalanced search tree
- **Goal: uniform** sampling of occurrences



Subgraph Occurrences on k -census are in the last tree level

Sampling approach

- **Original:** unbalanced search tree
 - **Goal:** **uniform** sampling



Associate a probability with traversing each search tree depth

Sampling approach

- **Probabilities associated with each depth:**

- $\{P_0, P_1, P_2, \dots, P_{\max}\}$

- **Sampling is uniform:**

- Probability of finding any occurrence is $P_0 \times P_1 \times P_2 \times \dots \times P_{\max}$

- **We can produce an unbiased estimator:**

- Estimate of frequency of subgraph $S =$

$$\frac{\text{Nr of sampled occurrences of } S}{P_0 \times P_1 \times P_2 \times \dots \times P_{\max}}$$

Sampling approach

- **The probabilities P_i control the search**
- **Regarding accuracy: avoid small values of probability close to the root**
 - Entire search branches disregarded → more variance
- **Regarding execution times: avoid high values if probability close to the root**
 - More search branches explored → more time
- **Choice should be balanced**

Sampling approach: some results

- **90% accuracy for motif detection in less than 20% of time**

[Ribeiro & Silva, WABI'2010]

- **First sampling process for customized sets of subgraphs**
 - Only sample the subgraphs we want
- **Many parametrization choices**
 - Adaptable for different use cases
 - Possible to refine prediction for desired set of subgraphs

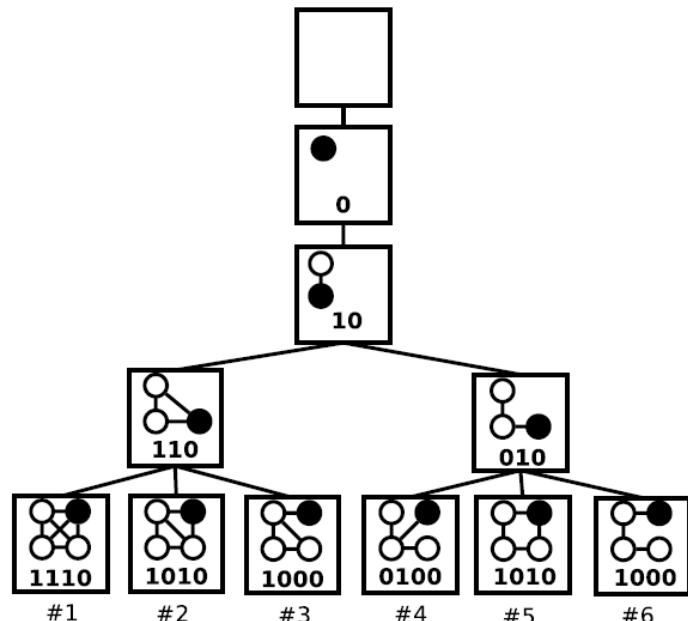
Adaptive sampling: ongoing work

- **Adapt the sampling process:**
 - To the network
 - To the subgraphs being searched
 - To the available running time
- **High level ideas of the algorithm:**
 - Do several sampling iterations and look at how estimations are converging
 - Ex: frequent subgraphs are easier to estimate
 - Change sampling weights
 - Changesubgraphs in the g-trie

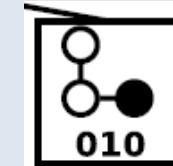
5) PARALLEL APPROACH

Opportunities for parallelization

- Sequential version produces a tree-shaped search tree
- Search tree nodes are independent from each other



{0,1,3}



If we know where we are,
we can continue from there

Tree Nodes -> Work Units

Initial Parallel Problem

- **Input:** set of **work units**
 - G-Trie: (Network, G-Trie Node, Partial Match)
 - ESU: (Network, Partial Match, Possible Extensions)
- **Goal:** **efficiently distribute** work units among processors
- **Initial target:** **distributed memory** with **message passing**

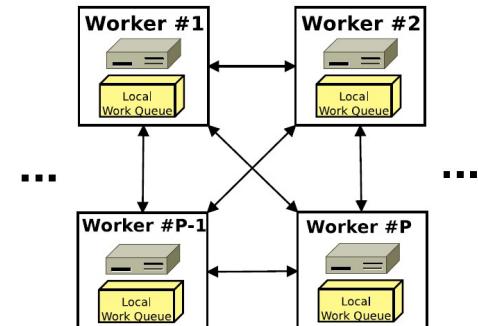
[Ribeiro, Silva & Lopes, Cluster'2010]
- **Constraints:** Tree highly **unbalanced**
 - Pre-determined static allocation is very hard!
 - Requires **dynamic load balancing**

Distributed Snapshot

Receiver-Initiated Strategy

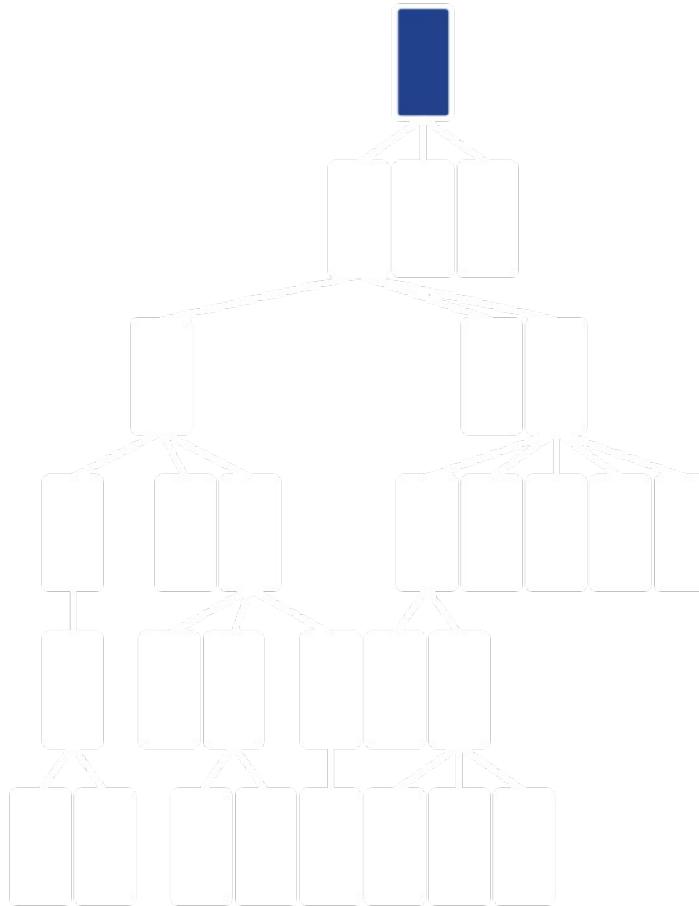
1) While computation not ended

- If work units available
 - . Process work unit
 - **Someone asked for work?**
 - > Stop my computation
 - > Divide work in 2 similar halves
 - > Send half to requester
 - > Return to computation
 - Else
 - Request work units from other processor



Running Computation

Example Computation



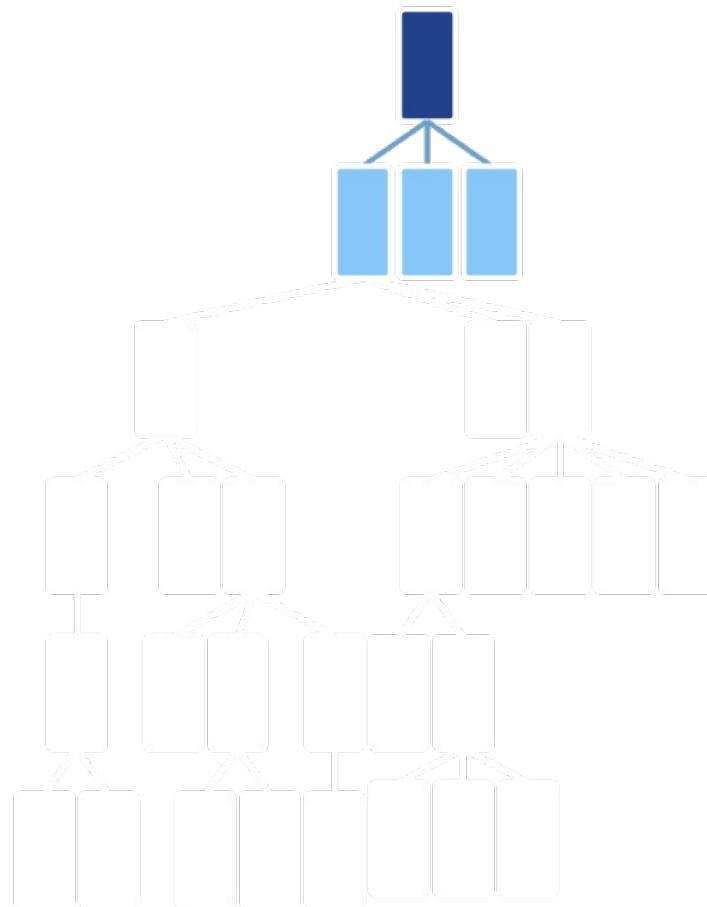
G-Trie Node



Graph Vertex

Running Computation

Example Computation

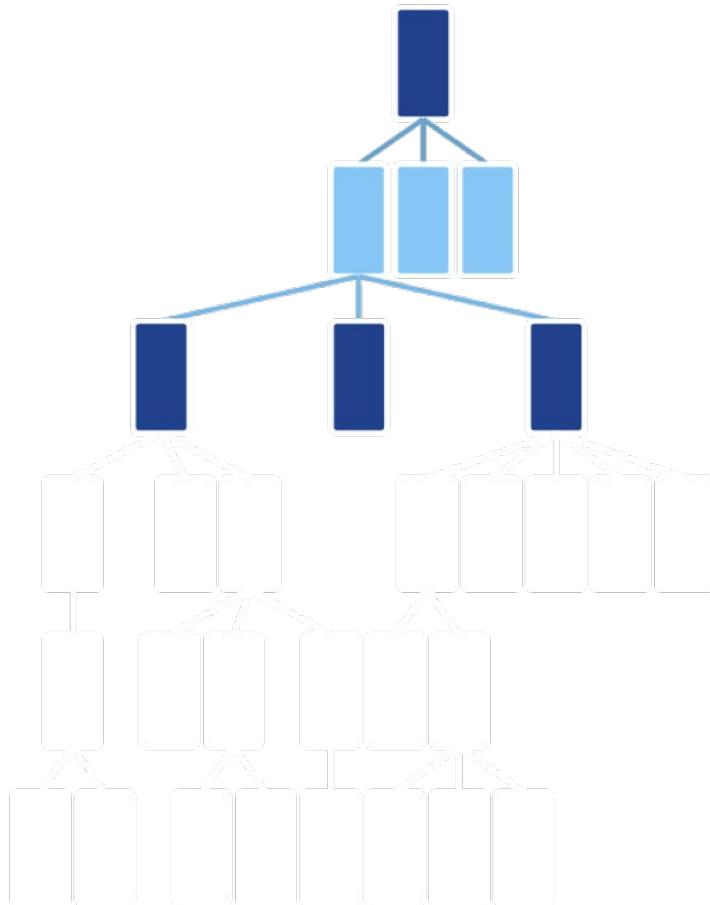


■ G-Trie Node
■ Graph Vertex

Running Computation

Example Computation

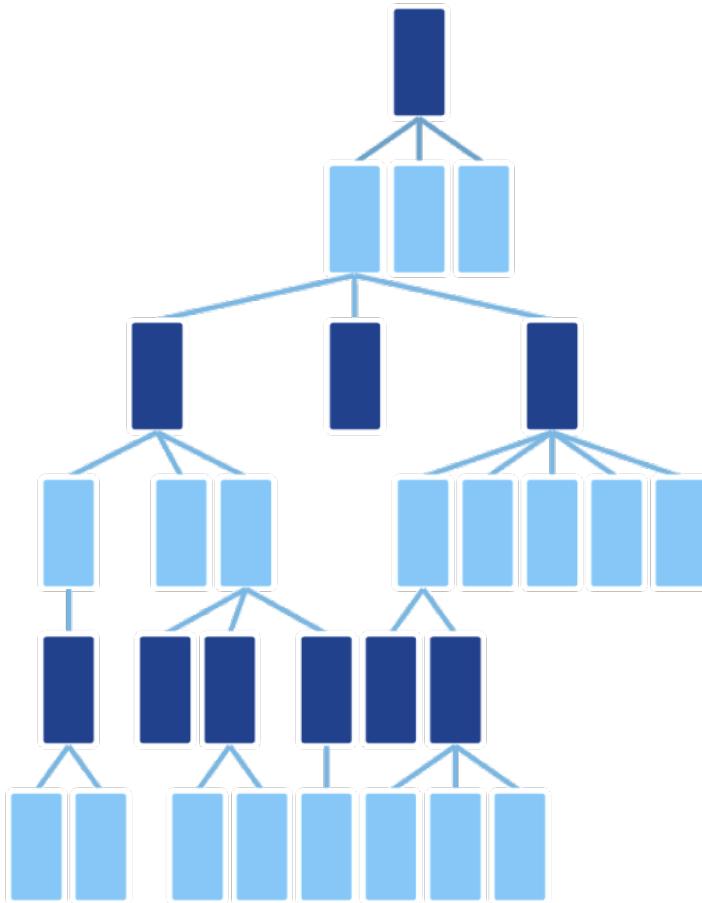
■ G-Trie Node
■ Graph Vertex



Running Computation

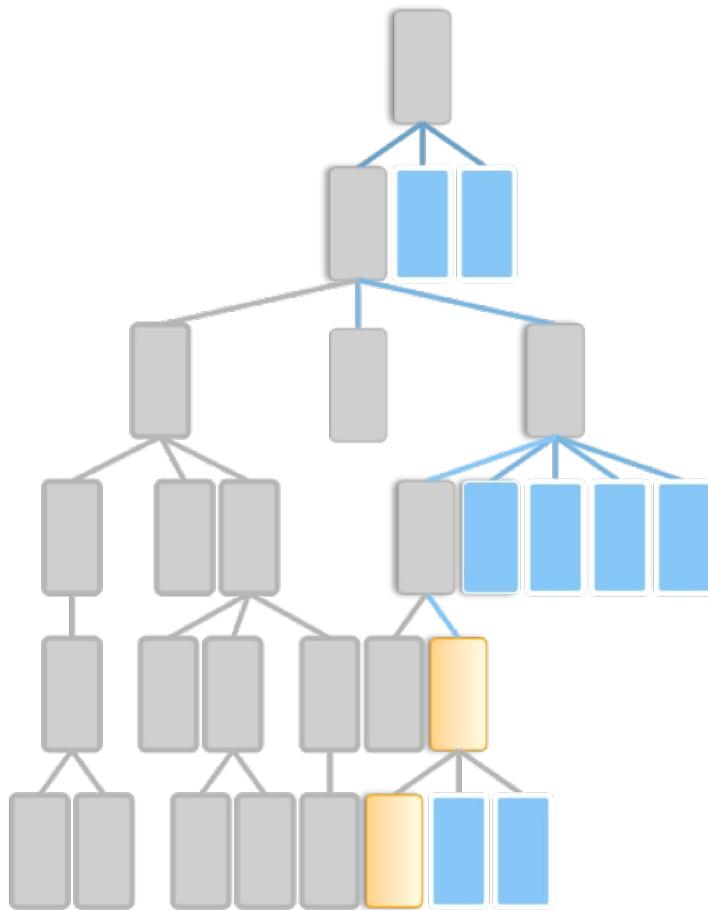
Example Computation

■ G-Trie Node
■ Graph Vertex



Stopping Computation

Example Computation



G-Trie Node

Graph Vertex

Current Work Unit

Explored Work Units

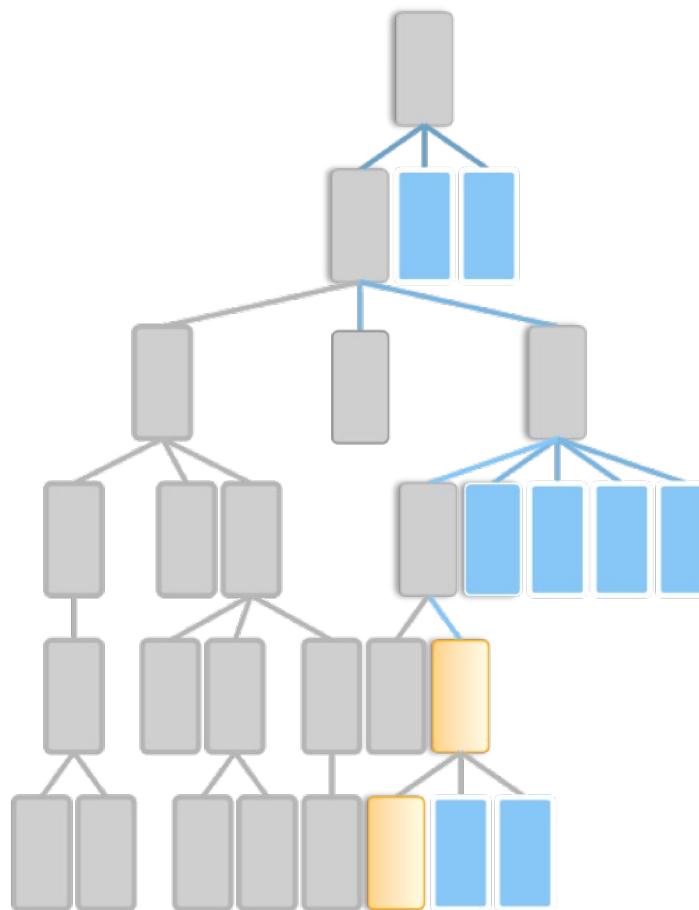


Dividing Computation

- **Goal:** divide work in **two “equal” halves**
- We create a **compact representation** of the search state (tree-shaped)
 - Take advantage of common substructure in work units
 - Efficient methods for: stopping, dividing, resuming
- We stop dividing when units are too small
 - Threshold in distance to search tree leaf
- We do a **diagonal split**
 - Round-robin scheme

Dividing Computation

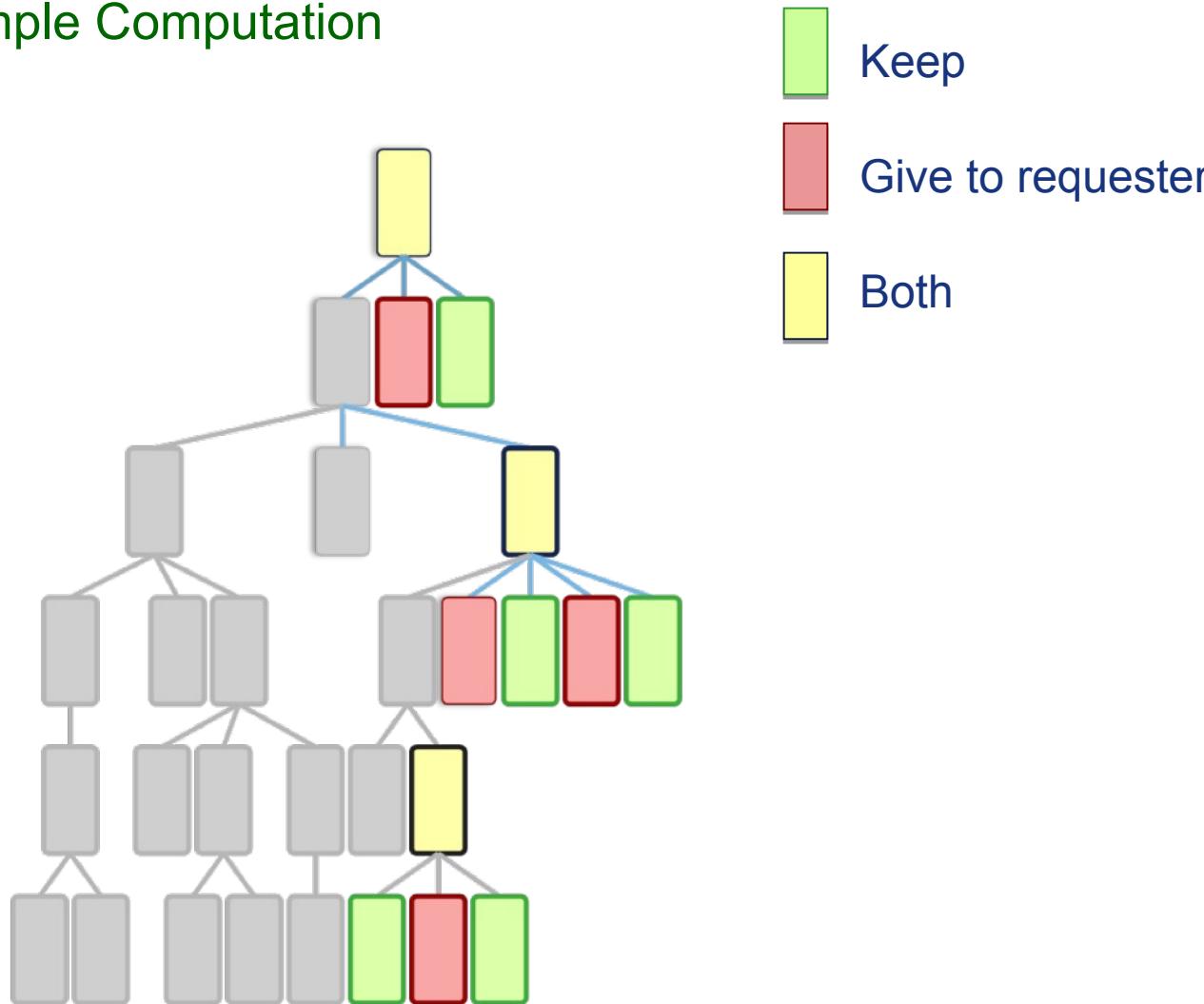
Example Computation



- █ G-Trie Node
- █ Graph Vertex
- █ Current Work Unit
- █ Explored Work Units

Dividing Computation

Example Computation



Work Request

- When we do not have work, which processor should we contact?

- No data locality
- Search trees completely unbalanced

- ✓ Ask a random processor!

- Random polling ([Sanders 1994])

Some Parallel Results

- **Absolute Speedup (*distributed snapshots*)**

Network	K	#CPUs: Speedup		
		32	64	128
dolphins	10	30.8	59.4	112.7
circuit	11	31.3	61.7	121.2
neural	6	31.4	62.5	122.8
metabolic	6	31.5	62.9	126.0
links	4	30.0	57.1	95.9
coauthors	8	31.4	62.6	123.9
ppi	6	31.4	62.0	122.1
odlis	4	29.7	55.9	90.2
power	9	31.1	61.0	118.8
company	5	31.3	62.8	125.2
foldoc	4	30.9	60.6	116.9
internet	4	31.4	62.9	125.7

Almost linear speedup up to 128 cores!

Some Parallel Results

[Aparício, Ribeiro & Silva, ISPA'2014]

- **Shared memory** implementation with similar results

Network	Subgraph size	#Subgraphs searched	Sequential time (s)	#Threads: speedup				Machine with 32 real cores			
				8	16	32	64	time (s)	8	16	32
polblogs	6	1,530,843	91,190.73	7.87	15.69	31.31	52.96	222,210.76	7.91	15.78	31.38
netsc	9	261,080	466.48	7.90	15.78	30.91	51.09	2,030.39	7.91	15.74	31.36
facebook	5	21	6,043.90	6.75	14.72	30.23	52.47	17,851.16	6.78	14.67	30.31
routes	5	21	4,936.54	6.53	14.52	30.34	48.76	20,706.67	6.80	14.67	30.53
company	6	1,530,843	26,955.71	6.74	14.54	29.99	45.12	94,384.39	6.69	14.61	30.17
blogcat	4	6	5,410.45	7.72	14.37	24.92	25.69	15,666.05	7.88	15.40	29.60
enron	4	199	1,038.60	6.23	12.69	23.78	24.41	2,768.74	6.42	13.69	27.43

G-Tries

Network	Subgraph size	#Leafs found	#Subgraph types found	Sequential time (s)	#Threads: speedup			
					8	16	32	64
jazz	6	3,113	112	295.95	6.75	14.86	29.92	49.74
polblogs	6	409,845	9,360	1,722.55	7.85	15.56	30.04	47.48
netsc	9	445,410	14,151	295.12	7.83	15.05	23.82	26.54
facebook	5	125	19	3,598.41	7.67	15.34	31.00	51.81
company	6	1,379	310	739.12	7.94	15.81	31.02	48.53
astroph	4	17	6	179.47	6.62	13.60	24.69	30.42
enron	4	17	6	1,370.46	7.70	13.32	25.44	35.85

FaSE

Almost linear speedup up to 32 cores!

Final Improvements

Combining:

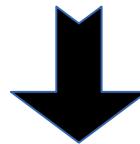
G-Trie Complete Sequential Improvement



Time Gains of Sampling Approach



Scalability of Parallel Approach



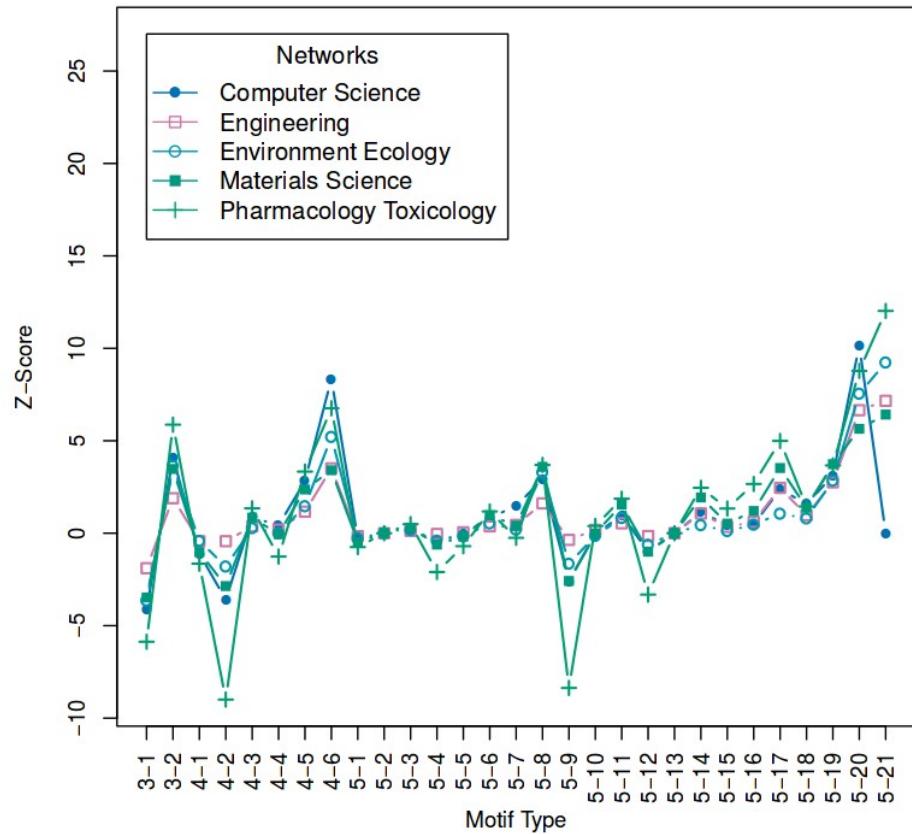
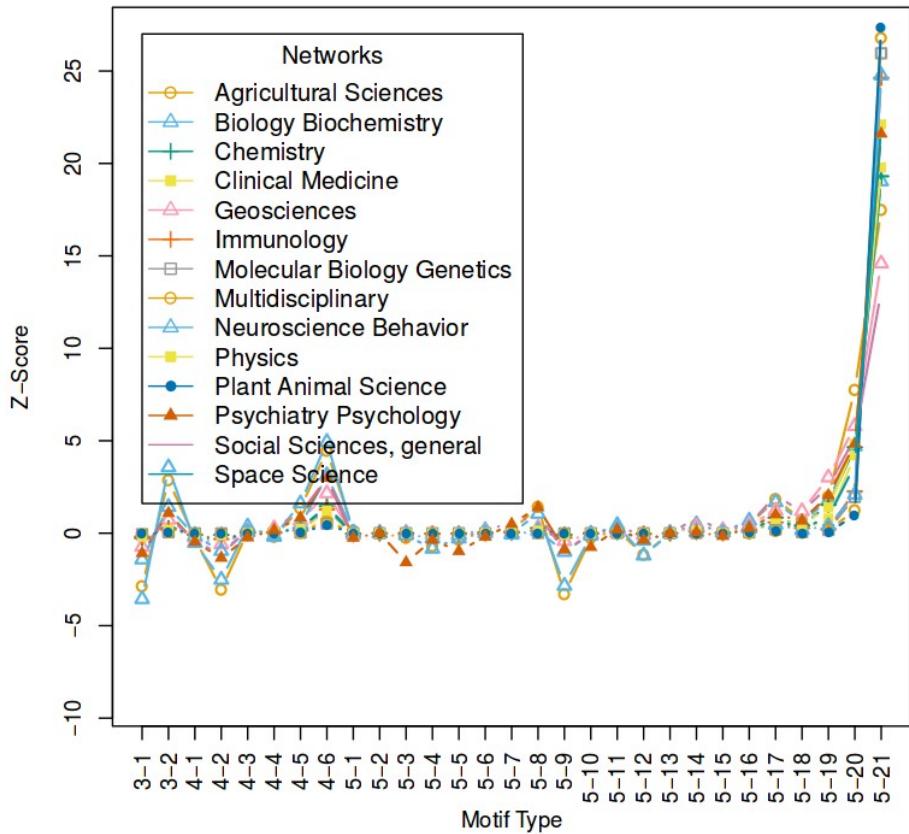
Over 2000x faster than previous state-of-the-art

Larger Networks
Larger Subgraphs
New Insight

6) EXAMPLE APPLICATIONS

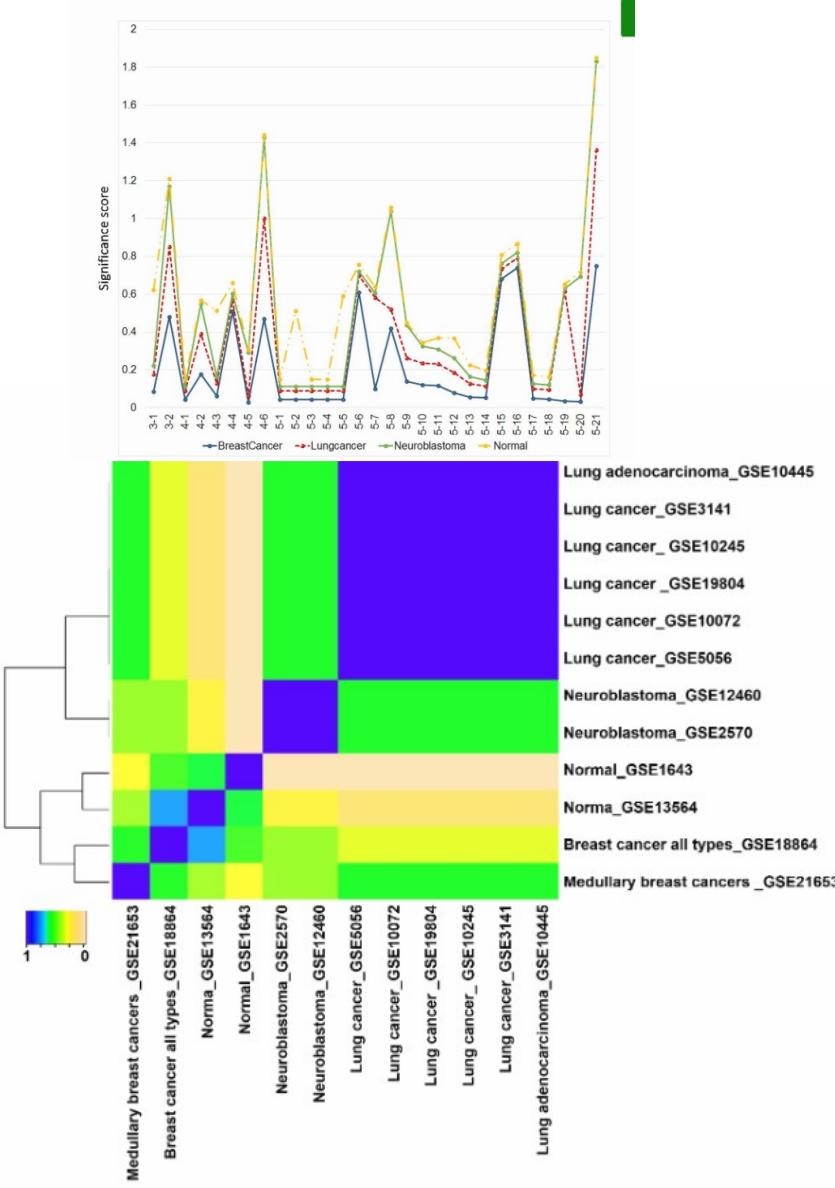
Co-Authorship Networks

Undirected Network Motifs



[Choobdar, Ribeiro & Silva,
ASONAM'2012]

Gene Co-Expression Networks



Weighted Network Motifs

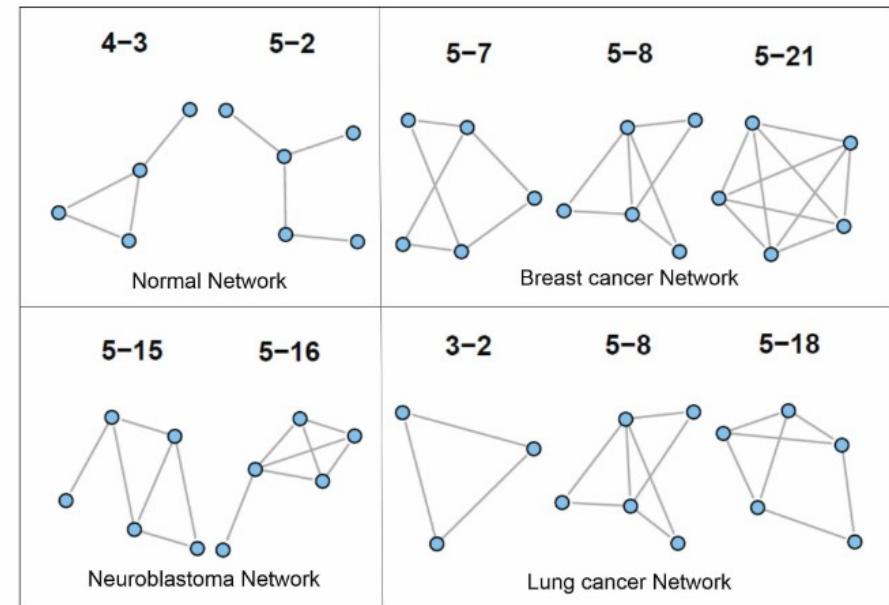
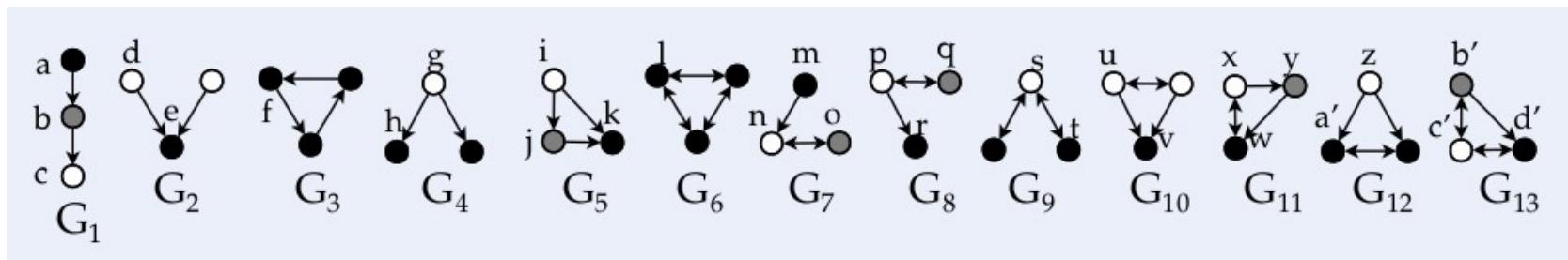
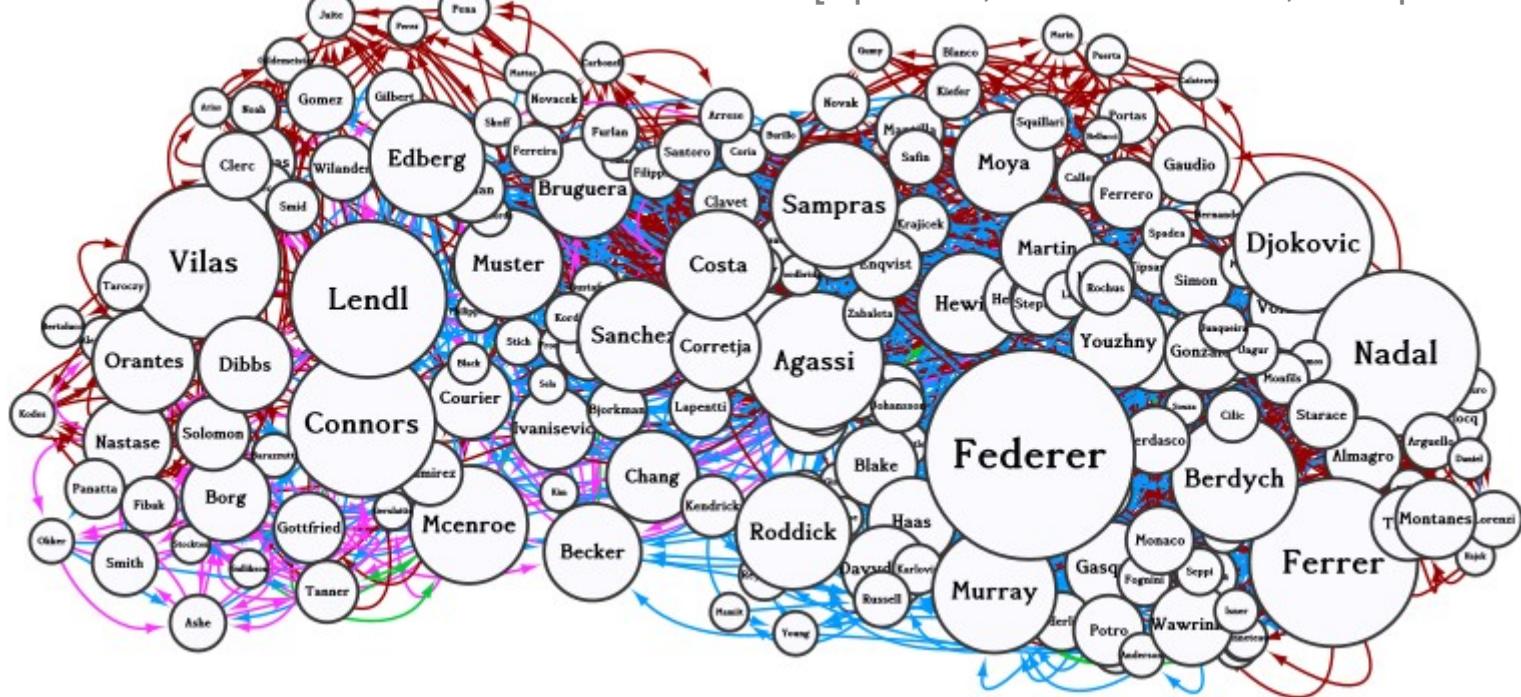


Figure 5: Discriminating subgraphs for each type of networks. [Cloud, Edge, Billing, S-GPU, SACGSS15]

[Choobdar, Ribeiro & Silva, SAC'2015]

Tennis Networks

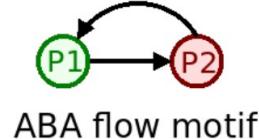
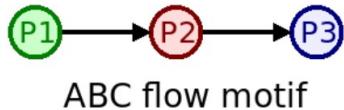
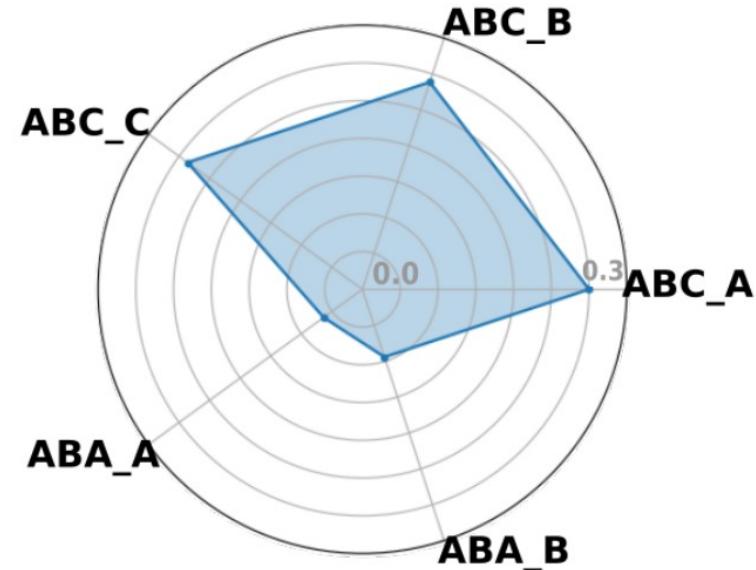
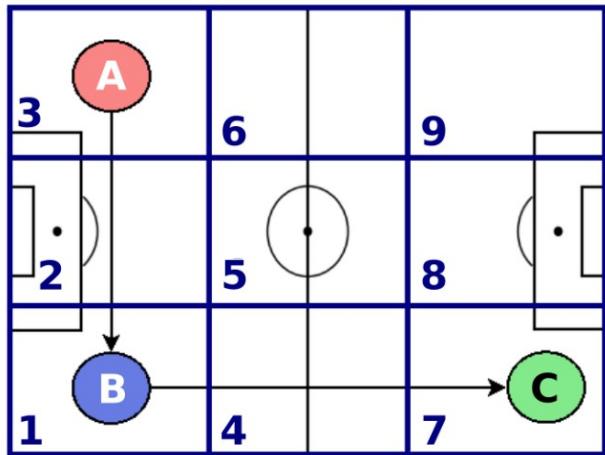
[Aparício, Ribeiro & Silva, Complenet'2016]



Dominance Patterns based on Directed Graphlets

Football Networks

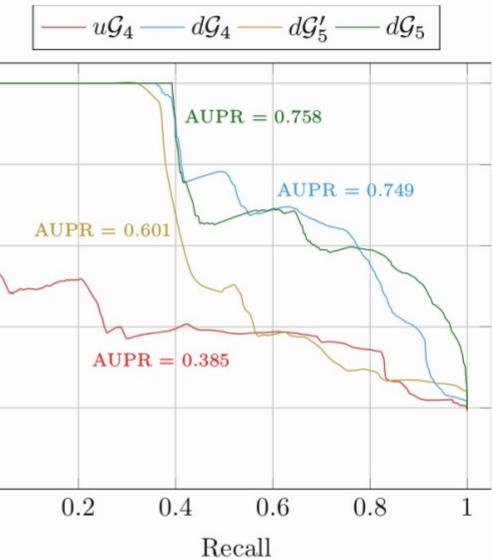
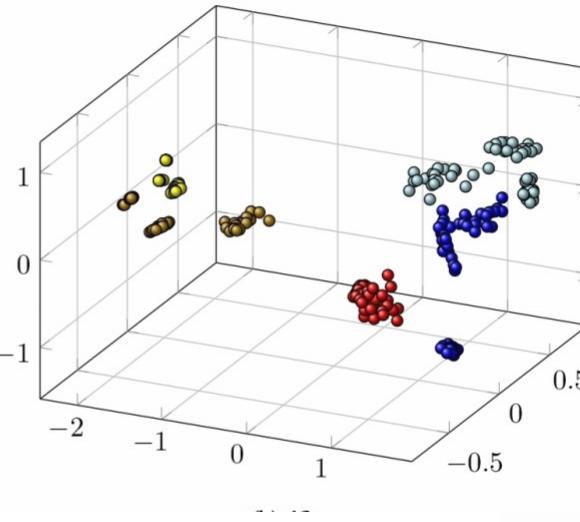
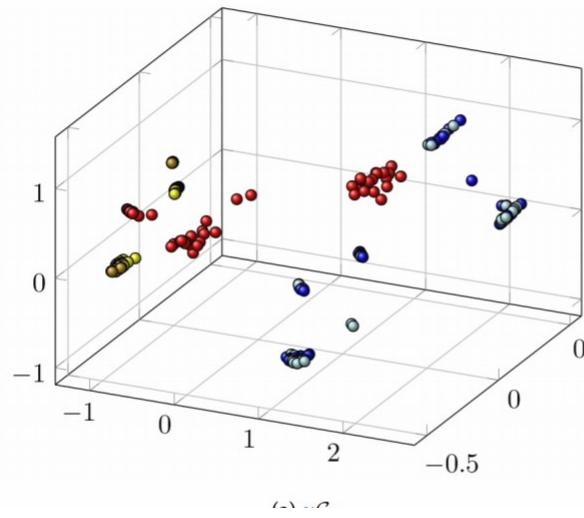
[Barbosa, Ribeiro and Dutra, CNA, 2022]



Flow Motifs in Passing Networks

Classifying and clustering

[Aparício, Ribeiro & Silva, TCBB, 2017]



Directed Graphlets

Classifying and clustering

[Aparício, Ribeiro & Silva, PLoS, 2018]

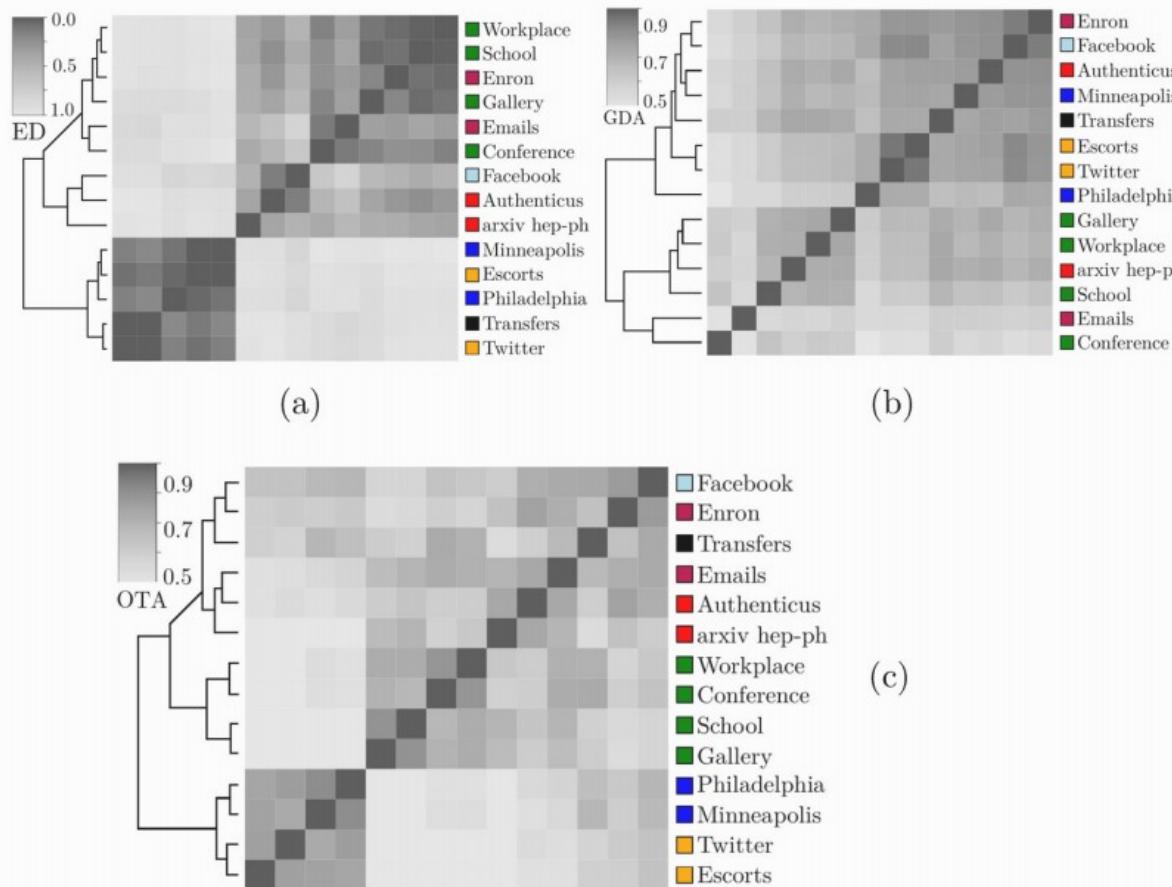


Fig 10. Similarity matrices according to (a) motif-fingerprints' Euclidean distance (ED), (b) graphlet-degree-agreement (GDA) and (c) orbit-transition-agreement (OTA). Clustering is performed using hierarchical clustering with complete linkage.

Graphlet-Orbit Transitions

Classifying and clustering

[Aparício, Ribeiro & Silva, PLoS, 2018]

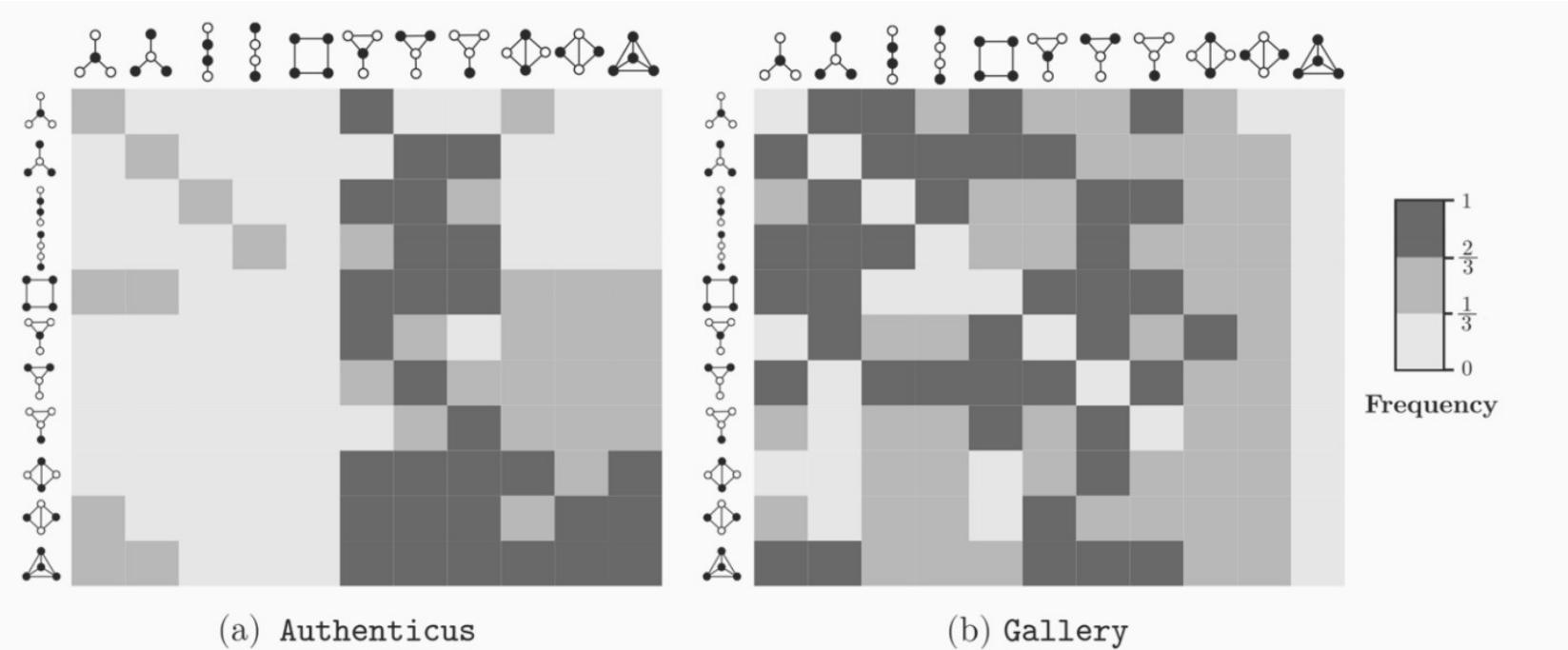


Fig 11. Orbit-transition matrices of (a) a collaboration network and a (b) physical interaction network for all 4-node orbits.

Graphlet-Orbit Transitions

7) RESOURCES

Some publications

Survey on existing Algorithms

- Survey on Subgraph Counting: Concepts, Algorithms, and Applications to Network Motifs and Graphlets.

ACM Computing Surveys, 2021.



Table 2. Overview of All Major Exact Algorithms

	Year	Approach	Type	k-restriction	Orbit	Directed	Code
MFINDER [122]	2002	Enum.	Classical	None	✗	✓	[9]
ESU [194, 197]	2005	Enum.	Classical	None	✗	✓	[195]
ITZHACK [71]	2007	Enum.	Classical	≤ 5	✗	✓	✗
GROCHOW [56]	2007	Enum.	Single-subgraph	None	✗	✓	✗
KAVOSH [79]	2009	Enum.	Classical	None	✗	✓	[123]
GTRIES [148, 150]	2010	Enum.	Encapsulation	None	✓	✓	[145]
RAGE [103, 104]	2010	Analytic	Decomposition	≤ 5	✗	✓	[105]
NeMo [86]	2011	Enum.	Single-subgraph	None	✗	✓	[156]
NETMODE [93]	2012	Enum.	Encapsulation	≤ 6	✗	✓	[94]
SCMD [186]	2012	Enum.	Encapsulation	None	✗	✗	✗
ACC-MOTIF [111, 112]	2012	Analytic	Decomposition	≤ 6	✗	✓	[110]
ISMAGS [40, 68]	2013	Enum.	Single-subgraph	None	✗	✓	[134]
QUATEXELERO [81]	2013	Enum.	Encapsulation	None	✗	✓	[82]
FASE [131]	2013	Enum.	Encapsulation	None	✗	✓	[146]
ENSA [206]	2014	Enum.	Encapsulation	None	✗	✓	✗
ORCA [62, 63]	2014	Analytic	Matrix-based	≤ 5	✓	✗	[64]
HASH-ESU [75]	2015	Enum.	Encapsulation	None	✗	✓	✗
SONG [177]	2015	Enum.	Encapsulation	None	✗	✓	✗
ORTMANN [128, 129]	2016	Analytic	Matrix-based	≤ 4	✓	✓	✗
PGD [3, 5]	2016	Analytic	Decomposition	≤ 4	✓	✗	[2]
PATCOMP [61]	2017	Enum.	Encapsulation	None	✗	✓	✗
ESCAPE [137]	2017	Analytic	Decomposition	≤ 5	✓	✗	[169]
JESSE [113, 115]	2017	Analytic	Matrix-based	None	✓	✗	[114]

Table 3. Algorithms for Approximate Subgraph Counting

	Year	Output	k-restriction	Directed	Strategy	Code
ESA [80]	2004	Conc.	None	✓	Random Walk	[9]
RAND-ESU [194]	2005	Freq.	None	✓	Rand. Enum.	[195]
TNP [140]	2006	Conc.	5	✗	Enum. - Generalize	✗
RAND-GTRIE [147]	2010	Freq.	None	✓	Rand. Enum.	[145]
GUISE [19]	2012	Conc.	5	✗	Random Walk	[142]
RAND-SCMD [186]	2012	Freq.	None	✓	Enum. - Generalize	✗
WEDGE SAMPLING [170]	2013	Freq.	3	✓	Path Sampling	[85]
GRAFT [143]	2014	Freq.	5	✗	Enum. - Generalize	[141]
PSRW & MSS [188]	2014	Conc.	None	✗	Random Walk	✗
MHRW [160]	2015	Conc.	None	✗	Random Walk	✓
RAND-FASE [132]	2015	Freq.	None	✓	Rand. Enum.	[133]
PATH SAMPLING [73]	2015	Freq.	4	✗	Path Sampling	✗
k-PROFILE SPARSIFIER [45, 46]	2016	Freq.	4	✗	Enum. - Generalize	[44]
MOSS [190]	2018	Freq.	5	✗	Path Sampling	[187]
SSRW [204]	2018	Freq.	7	✗	Random Walk	✗
CC [25]	2018	Freq.	None	✗	Color Coding	[24]

Table 4. Algorithms for Approximate Subgraph Counting with Restricted Access

	Year	Output	k-restriction	Directed	Strategy	Code
WRW [59]	2016	Conc.	None	✗	Random Walk	✗
IMPR [31]	2016	Freq.	5	✗	Random Walk	[29]
CSS & NB-SRW [30]	2016	Conc.	None	✗	Random Walk	✓
MINFER [189]	2017	Conc.	5	✓	Enumerate - Generalize	✗

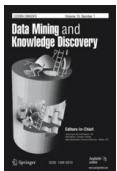


- Strategies for Network Motifs Discovery. E-Science 2009.

Some publications

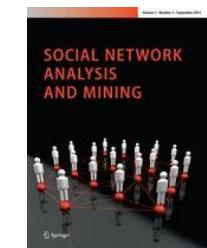
Core complete sequential algorithms

- Large Scale Graph Representations for Subgraph Census. NetSciX'2016
- G-Tries: a data structure for storing and finding subgraphs. Data Mining and Know. Discovery, 2014.
- Towards a faster network-centric subgraph census. ASONAM'2013
- Querying Subgraph Sets with G-Tries. DBSocial'2012 (best paper award)



Sampling approach

- Rand-Fase: Fast Approximate Subgraph Census. SNAM'2015.
- Efficient Subgraph Frequency Estimation with G-Tries. WABI'2010.



Parallel approach

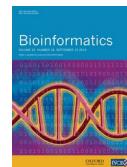
- Scalable Subgraph Counting using MapReduce. ACM'SAC 2017
- Parallel subgraph counting for multicore architectures. ISPA'2014
- A Scalable Parallel Approach for Subgraph Census Computation. MuCoCos'2014
- Parallel Discovery of Network Motifs. Journal of Parallel and Distributed Computing. 2012.
- Efficient Parallel Subgraph Counting using G-Tries. IEEE Cluster'2010.



Some publications

Concept variations and applications

- Computing Motifs in Hypergraphs. CompleNet, 2024
- Improving the Characterization and Comparison of Football Players with Spatial Flow Motifs. CNA, 2022
- Towards the Concept of Spatial Network Motifs. CNA, 2022
- Condensed Graphs: A Generic Framework for Accelerating Subgraph Census Computation. CompleNet, 2020
- Streamfase: An online algorithm for subgraph counting in dynamic networks. CNA, 2020
- Finding Dominant Nodes Using Graphlets. CNA, 2019
- Temporal network alignment via GoT-WAVE. Bioinformatics, 2019
- Graphlet-orbit Transitions (GoT): A fingerprint for temporal network comparison. PloS One, 2018
- Fast streaming small graph canonization. CompleNet'2018
- Network motifs detection using random networks with prescribed subgraph frequencies. CompleNet'2017
- Extending the applicability of Graphlets to Directed Networks. T C Biology and Bioinformatics, 2016
- A subgraph-based ranking system for professional tennis players. CompleNet'2016
- Discovering weighted motifs in gene co-expression networks. ACM-SAC'2015
- Discovering Colored Network Motifs. CompleNet'2014
- Co-authorship network comparison across research fields using motifs. ASONAM'2012.
- Motif Mining in Weighted Networks. Damnet'2012



Software

- **Reference sequential implementation** (C++)

<http://www.dcc.fc.up.pt/~prieiro/gtries/>

- **Parallel Implementation** (C++ pthreads, multicores)

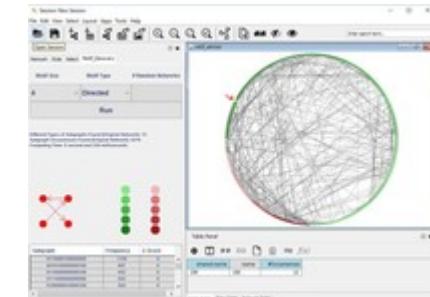
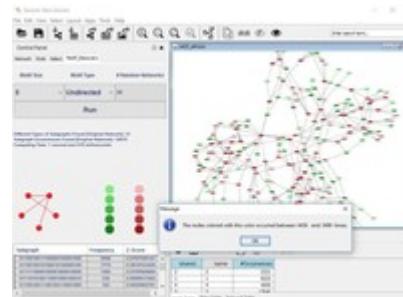
<http://www.dcc.fc.up.pt/~daparicio/software.html>

- **Cytoscape App** (Java, “alpha” version)

<http://apps.cytoscape.org/apps/motifdiscovery>



Motif Analysis Results				
Subgraph	Org. Frequency	Z-score	Rnd. Frequency	
0111 0000 0000 0000	148761	0.00	0.00 +/- 0.00	
0000 0001 1001 1000 0000	22995	0.00	0.00 +/- 0.00	
0010 0101 0001 0000	4498	0.00	0.00 +/- 0.00	
0110 0000 0000 0110	1843	0.00	0.00 +/- 0.00	
0011				



Master's Thesis

Many of my **MSc students** did their **thesis** centered on **subgraphs** and/or **motifs**.
For instance:

- **Studying GNNs and their Capabilities for Finding Motifs.** Pedro Vieira, M:DS, 23/24
- **Subgraph Patterns in Spatial Networks.** José Ferreira, M:CC, 21/22
- **Subgraph Patterns in Colored Networks.** Beatriz Pinto, M:CC, 20/21
- **Counting Subgraphs in Streaming Networks.** Henrique Branquinho, M:CC, 19/20
- **From Supergraph Counting to Subgraph Generation.** Luciano Grácio, M:CC, 18/19
- **Subgraph Patterns in Multiplex Networks.** André Meira, MI:ERSI, 18/19
- **Condensed Graphs: Towards a General Approach for Faster Subgraph Census.** Miguel Martins, M:CC, 18/19
- **Motif Based Community Discovery.** Rui Fonseca, M:CC, 17/18
- **Adaptive Parallel Subgraph Sampling in Large Complex Networks.** André Cascais, M:CC, 17/18
- **Counting subgraphs: from static to dynamic networks.** Name, M:CC, 16/17
- **Large Scale Parallel Subgraph Search.** Ahmad Naser Eddin, M:CC, 15/16
- **Pattern Discovery in Complex Network using Parallelism.** David Aparício, M:CC, 13/14



PhD Thesis

My own PhD was centered on algorithms for motif discovery:

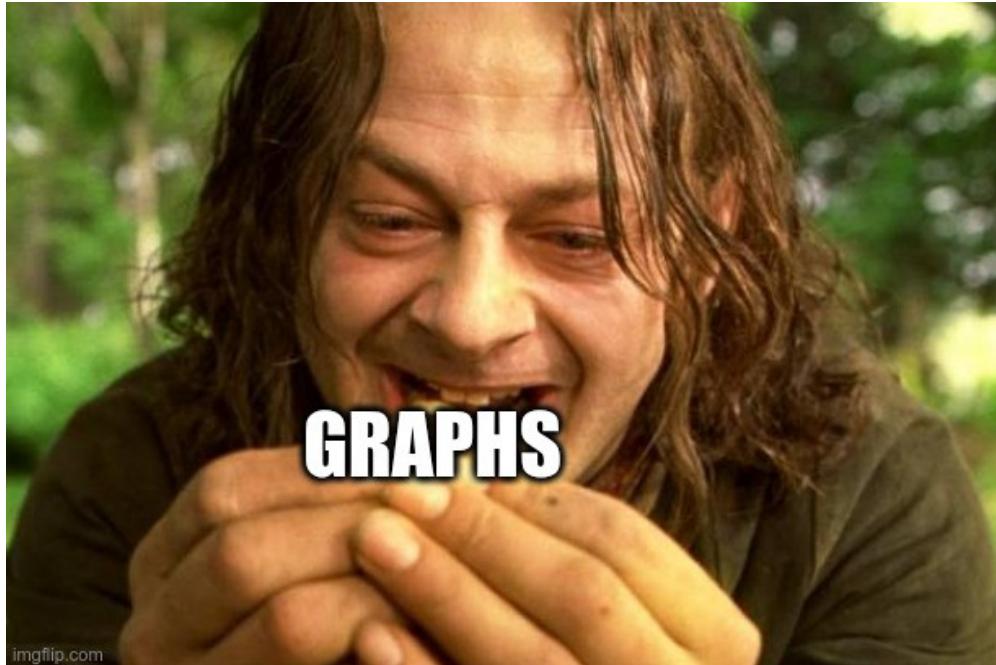
PhD Thesis (2011)

Efficient and Scalable Algorithms for Network Motifs Discovery



Network Science 24/25

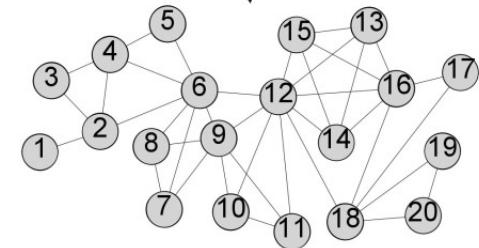
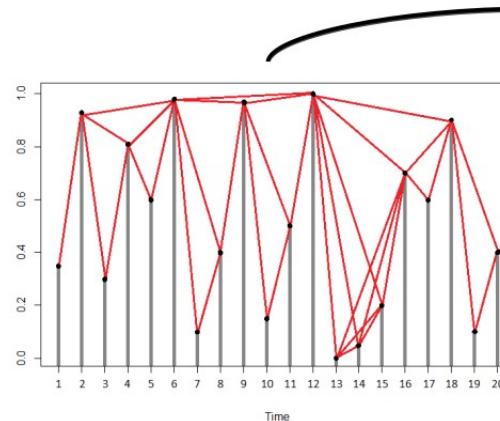
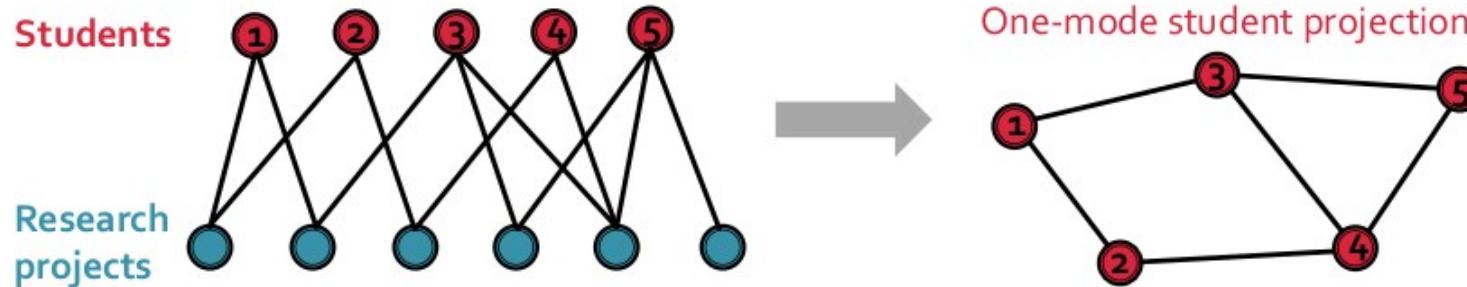
How it started...



How it's going!

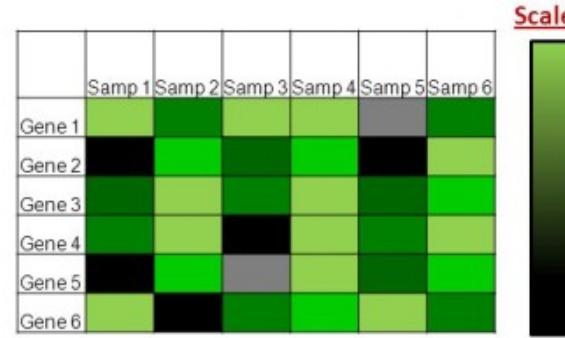
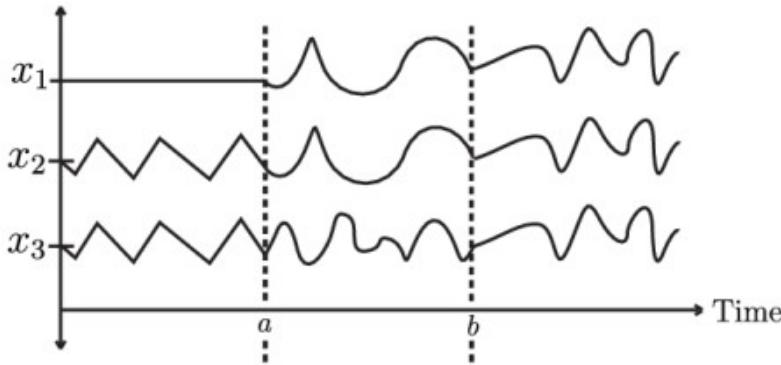


Network Construction



(Mainly selected slides from Jure Leskovec, Lucas Lacasa and Vanessa Silva)

Raw data is often NOT a network



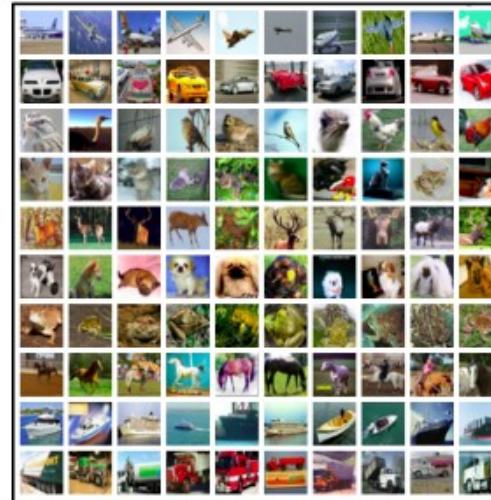
```
[+] Affable_Nitwit 559 points 15 hours ago
NOW That's What I Call Music. We're on 53, everybody. I checked. Online.
permalink save report give gold reply

[+] AggressiveToothbrush 165 points 15 hours ago
For the record, as of May 4th, we're on 54.
http://en.wikipedia.org/wiki/Now\_That%27s\_What\_I\_Call\_Music!\_54\_%28U.S.\_series%29
permalink save parent report give gold reply

[+] summertorother 224 points 15 hours ago
Speak for yourselves, the UK is on 90.
permalink save parent report give gold reply

[+] anideaguy 206 points 10 hours ago
TIL 54 is 90 in metric.
permalink save parent report give gold reply

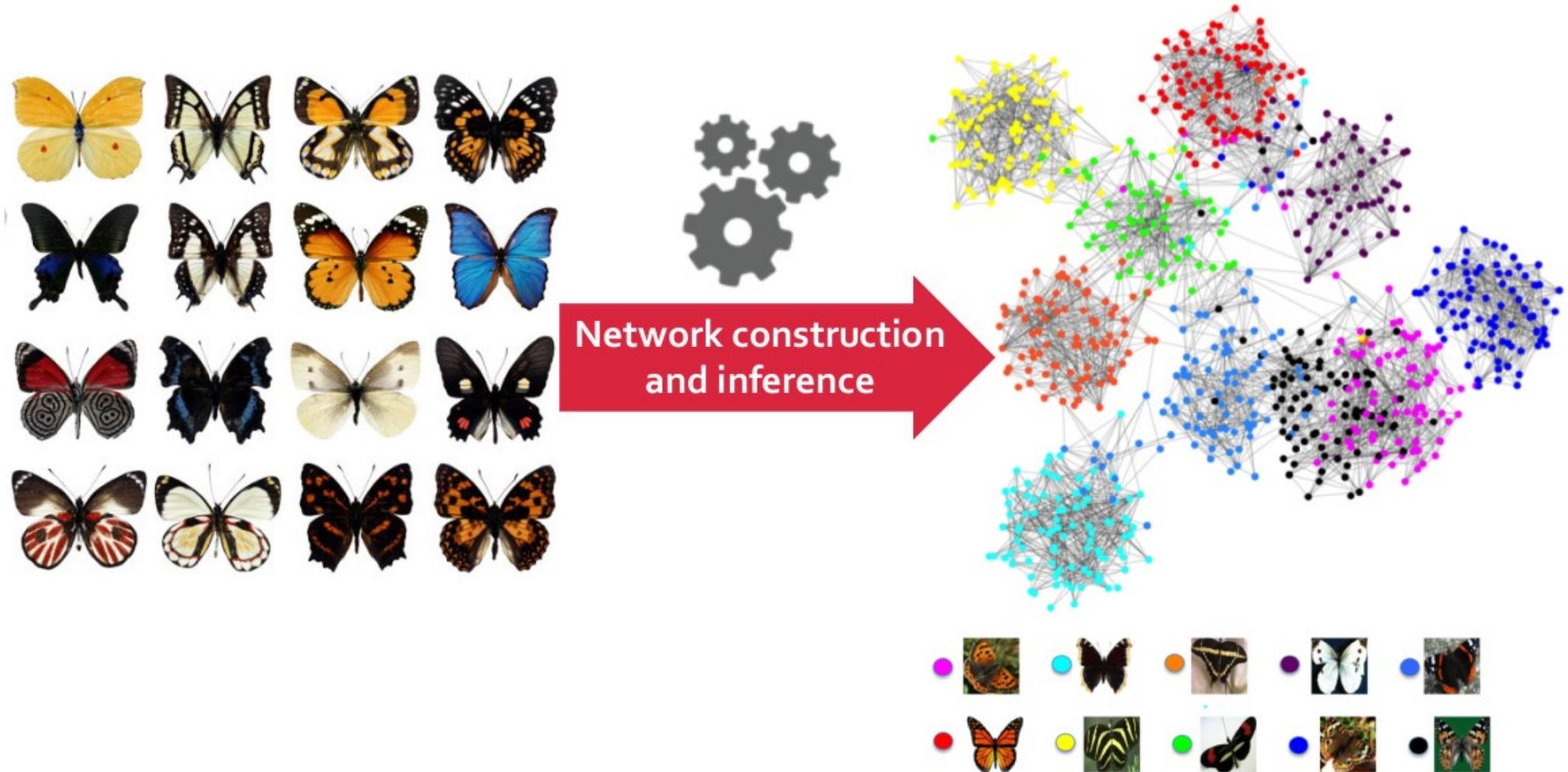
[+] LumberCockSucker 11 points 3 hours ago
The UK uses imperial measurements
permalink save parent report give gold reply
```



	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Feature matrices, relationship tables, time series, document corpora, image datasets, etc.

How to construct networks?



**Today: How to construct and infer networks
from raw data?**

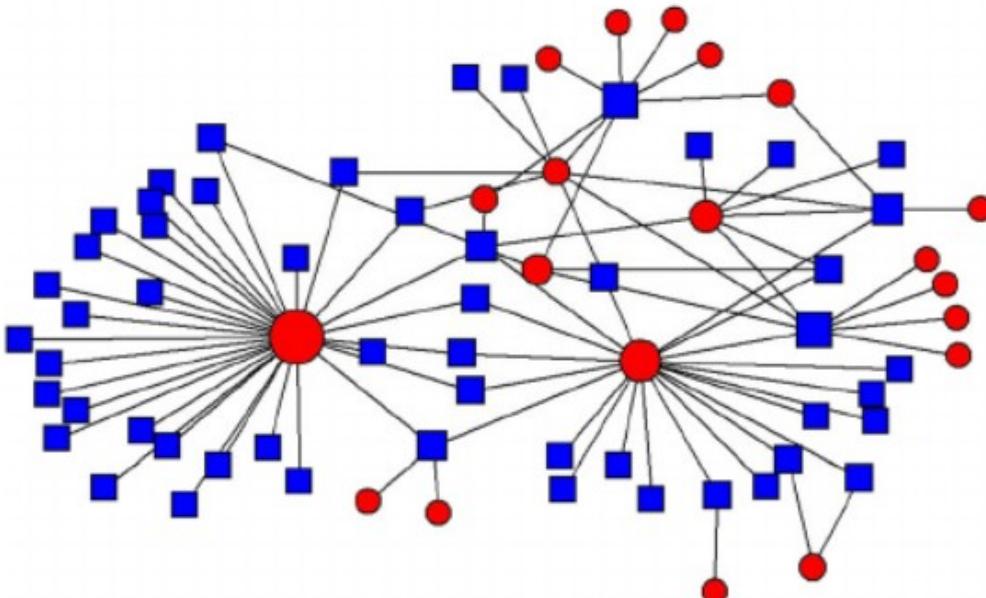
Plan for Today

- **Multi-mode network transformations**
 - K-partite graphs and projections
 - Graph Contractions
- **K-nearest neighbor graphs**
- **Network deconvolution**
 - Direct and indirect effects
- **From time-series to graphs**
 - Visibility and quantile graphs

Multi-Mode Network Transformations

Bipartite and K-partite Networks

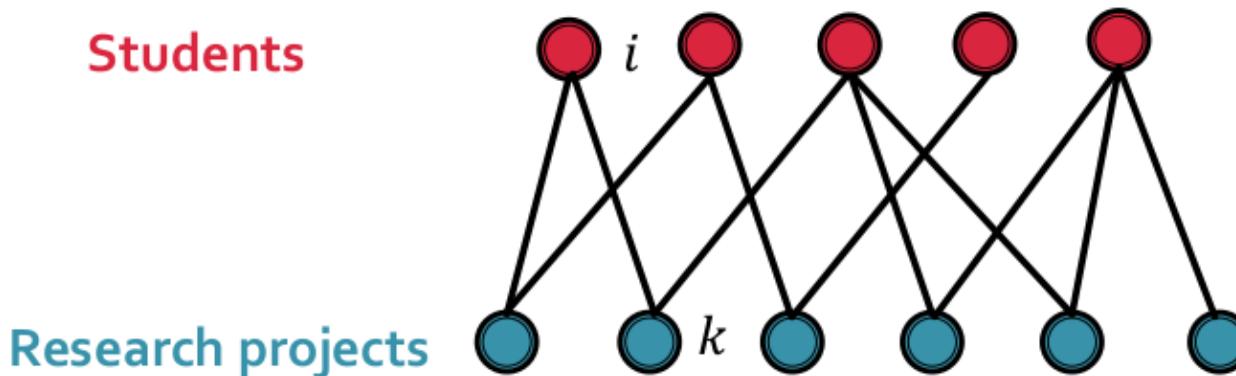
- Most of the time, when we create a network, all nodes represent **objects of the same type**:
 - People in social nets, bus stops in route nets, genes in gene nets
- **Multi-partite networks** have **multiple types of nodes**, where edges exclusively go from one type to the other:
 - **2-partite student net:** Students <-> Research projects
 - **3-partite movie net:** Actors <-> Movies <-> Movie Companies



Network on the left is a social bipartite network. **Blue squares** stand for people and **red circles** represent organizations

One-mode Projections: Example

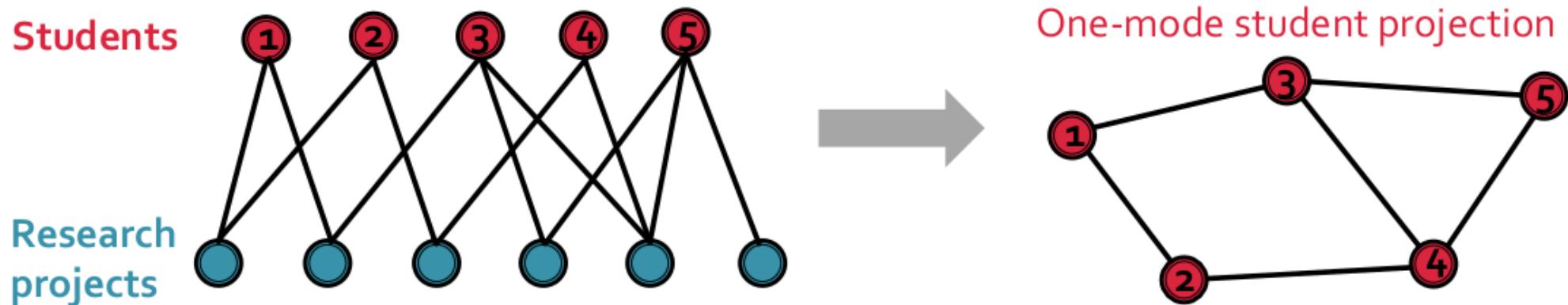
- **Example:** Bipartite student-project network:
 - **Edge:** Student i works on research project k



- **Two network projections of student-project network:**
 - **Student network:** Students are linked if they work together in **one or more projects**
 - **Project network:** Research projects are linked if **one or more students** work on both projects
- **In general:** K-partite network has K one-mode network projections

One-mode Projections: Example

- **Example:** Projection of bipartite student-project network onto the **student mode**:

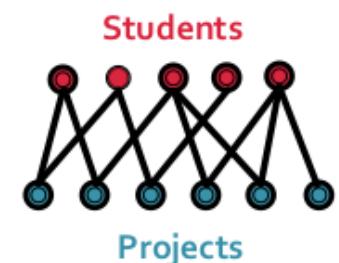


- Consider students 3, 4, and 5 connected in a triangle:
 - Triangle can be a result of:
 - **Scenario #1:** Each pair of students work on a different project
 - **Scenario #2:** Three students work on the same project
 - One-mode network projections **discard some information:**
 - Cannot distinguish between #1 and #2 just by looking at the projection

Constructing One-mode Projections

- **One-mode projection onto student mode:**
 - #(projects) that students i and j work together on is equivalent to the number of **paths of length 2** connecting i and j in the bipartite network
- Let C be **incidence matrix** of student-project net:

$$C_{ik} = \begin{cases} 1 & \text{if } i \text{ works on project } k \\ 0 & \text{otherwise} \end{cases}$$

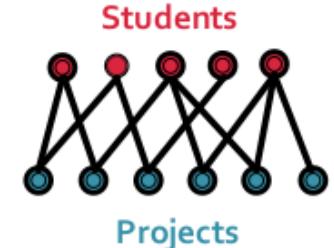


- C is an $n \times m$ **binary non-symmetric matrix**:
 - n is #(students), m is #(projects)

Constructing One-mode Projections

- **Idea:** Use C to construct various one-mode network projections
- **Weighted student network:**

$$B_{ij} = \begin{cases} w_{ij}, & \#(\text{projects}) \text{ that } i \text{ and } j \text{ collaborate on} \\ 0 & \text{otherwise} \end{cases}$$



- $B_{ij} = \sum_{k=1}^m C_{ik} C_{jk}$, i.e., the number of **paths of length 2** connecting students i and j in the bipartite network
- $\mathbf{B} = \mathbf{C}\mathbf{C}^T$ and B_{ii} represents $\#(\text{projects})$ that student i works on
- **Similarly, weighted project network:**

$$D_{kl} = \begin{cases} w_{kl}, & \#(\text{students}) \text{ that work on } k \text{ and } l \\ 0 & \text{otherwise} \end{cases}$$

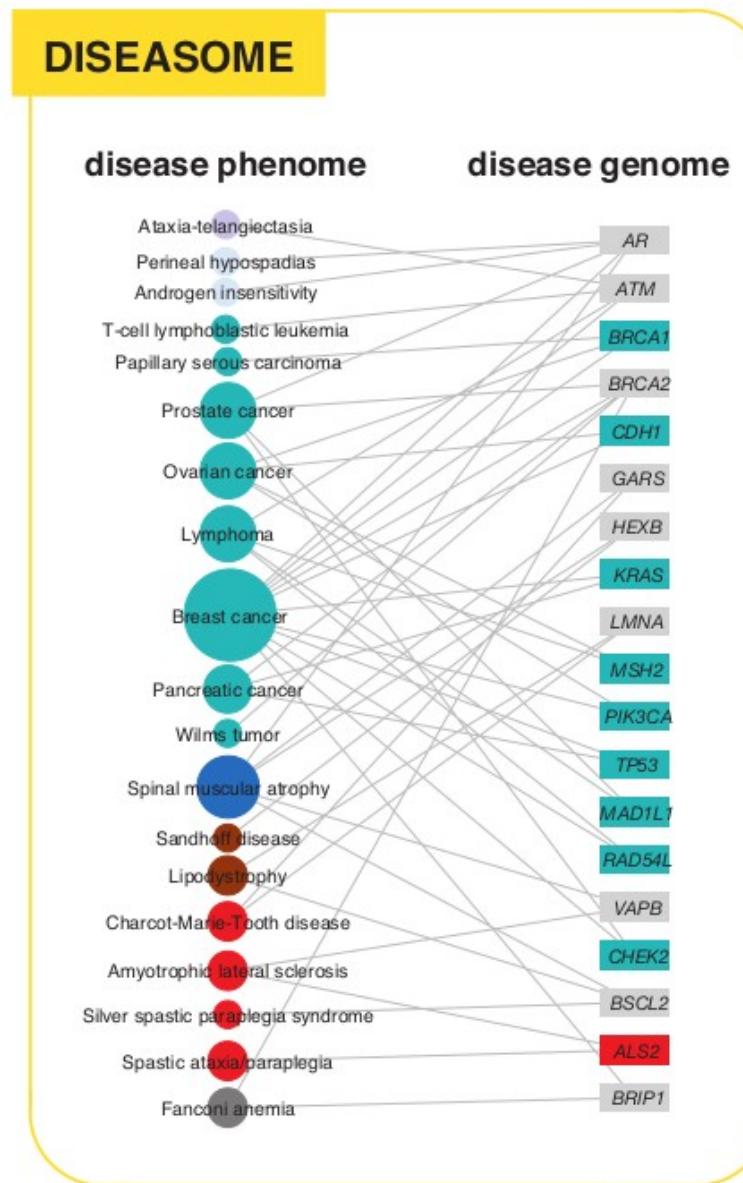
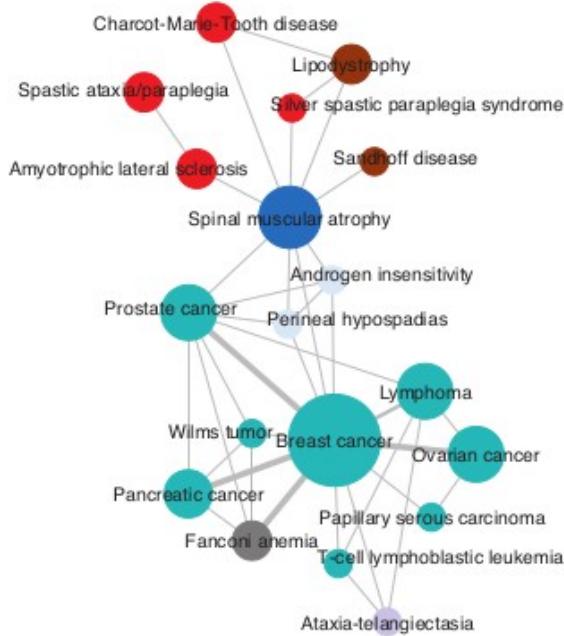
- $D_{kl} = \sum_{i=1}^n C_{ik} C_{il}$, i.e., the number of **paths of length 2** connecting projects k and l in the bipartite network
- $\mathbf{D} = \mathbf{C}^T \mathbf{C}$ and D_{kk} represents $\#(\text{students})$ that work on project k
- **Next:** Use \mathbf{B} and \mathbf{D} to obtain different network projections

Constructing One-mode Projections

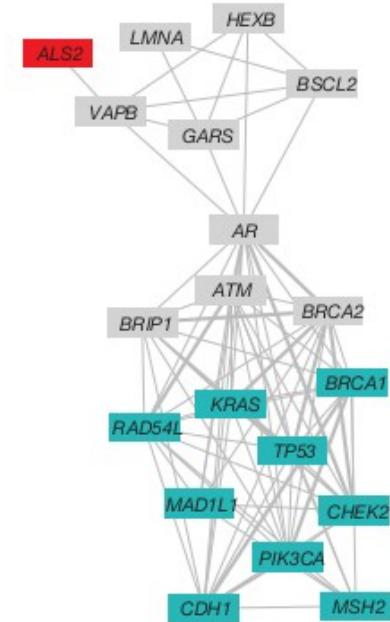
- Construct network projections by applying **a node similarity measure** to B and D
- **Two node similarity measures:**
 - **Common neighbors:** #(shared neighbors of nodes)
 - **Student network:** i and j are linked if they work together in **r or more projects, i.e.,** if $B_{ij} \geq r$
 - **Project network:** k and l are linked if **r or more students** work on both projects, *i.e.*, if $D_{kl} \geq r$
 - **Jaccard index:**
 - Common neighbors with a penalization for each non-shared neighbor:
 - Ratio of shared neighbors in the complete set of neighbors for 2 nodes
 - **Student network:** i and j are linked if they work together in **at least p fraction of their projects, i.e.,** if $B_{ij}/(B_{ii} + B_{jj} - B_{ij}) \geq p$
 - **Project network:** k and l are linked if **at least p fraction of their students** work on both projects, *i.e.*, if $D_{kl}/(D_{kk} + D_{ll} - D_{kl}) \geq p$

Example: Human Disease Network

Human Disease Network (HDN)



Disease Gene Network (DGN)

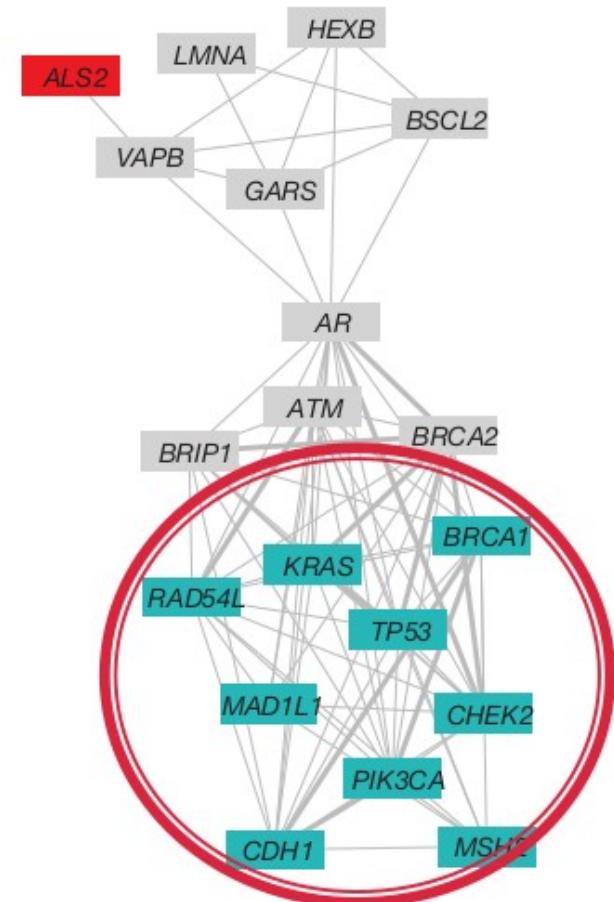


Kwang-II Goh *et al.*, The human disease network. *PNAS*, 104:21, 2007.

Example: Human Disease Network

- **Issue:** Folded gene network contains many **cliques**:
 - Why do cliques arise in the folded gene network?
 - Homework 1
- **Cliques make the network difficult to analyze:**
 - Computational complexity of many algorithms depends on the size and **number of large cliques**
- **Solution:** Use **graph contraction** to eliminate cliques

Disease Gene Network
(DGN)



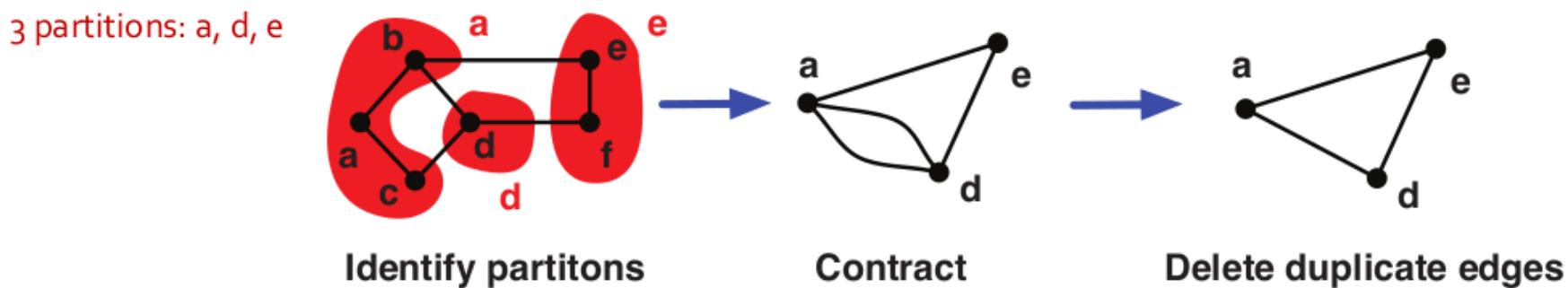
A clique of 9 gene nodes

Graph Contraction

- **Graph contraction:** Technique for computing properties of networks in parallel:
 - **Divide-and-conquer principle**
- **Idea:**
 - **Contract the graph** into a smaller graph, ideally a constant fraction smaller
 - Recurse on the smaller graph
 - Use the result from the recursion along with the initial graph to calculate the desired result
- **Next:** How to contract (“shrink”) a graph?

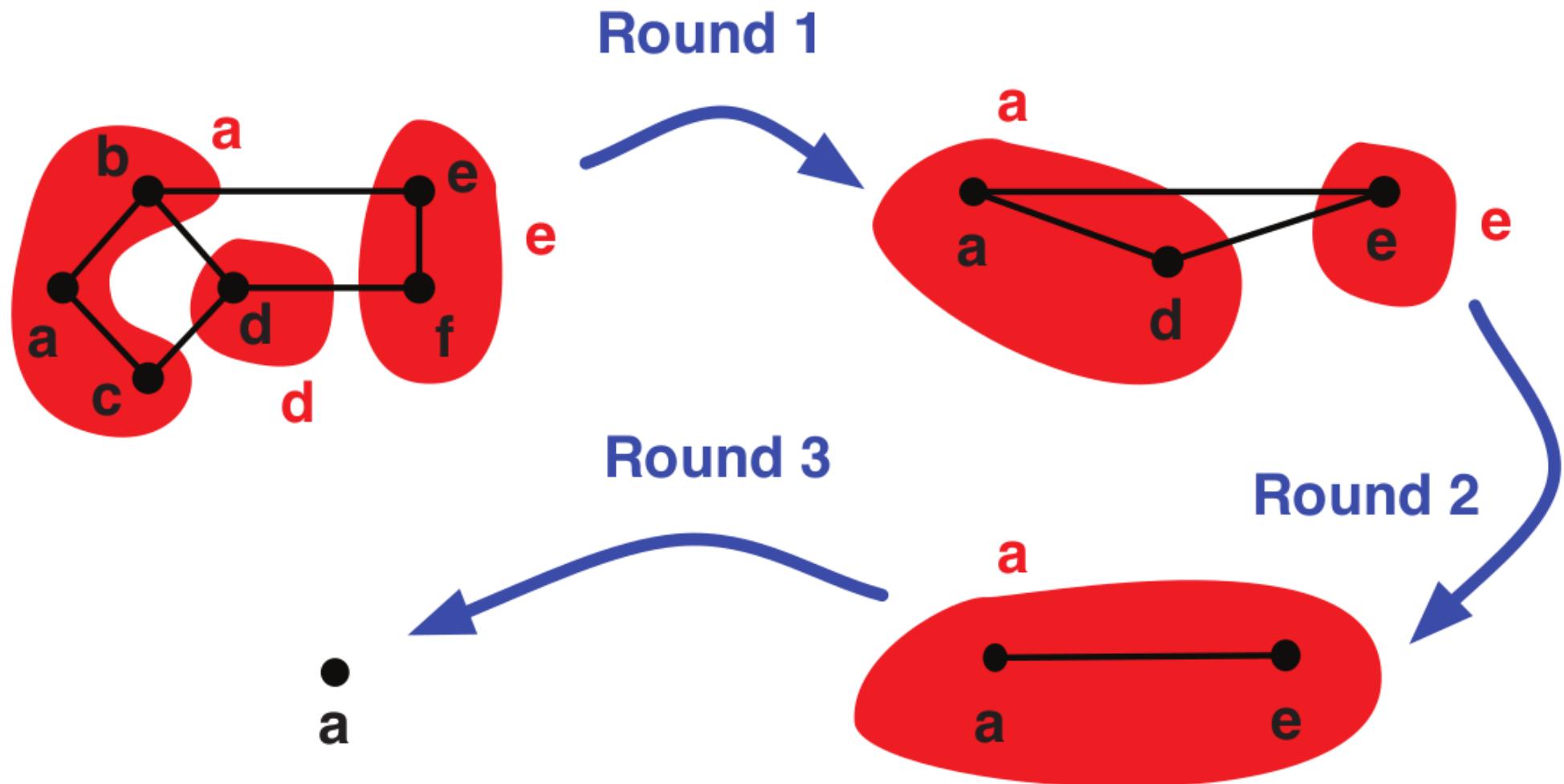
Graph Contraction: Algorithm

- Start with the input graph G :
 1. Select a **node-partitioning** of G to guide the contraction:
 - Partitions are disjoint and they include all nodes in G
 2. Contract each partition into a single node, a **supernode**
 3. Drop edges internal to a partition
 4. Reroute cross edges to corresponding **supernodes**
 5. Set G to be the smaller graph; Repeat
- Example: one round of graph contraction:



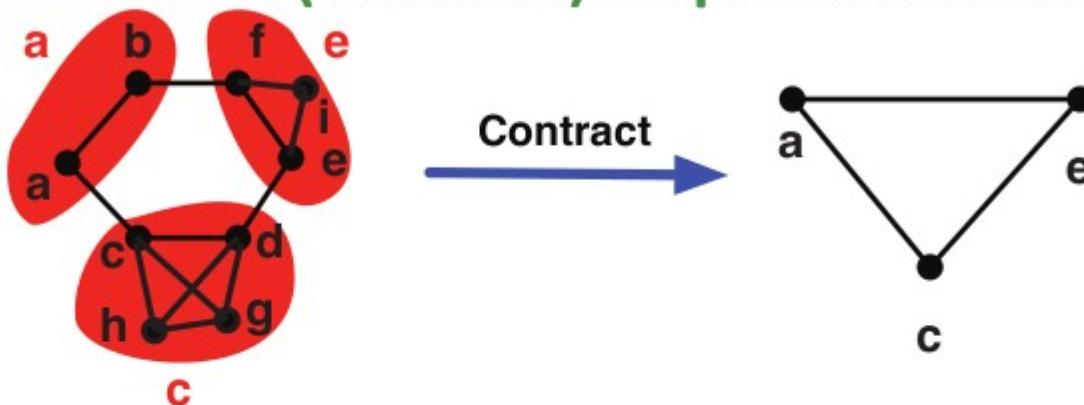
Graph Contraction: Example

Contracting a graph down to a single node in three rounds:

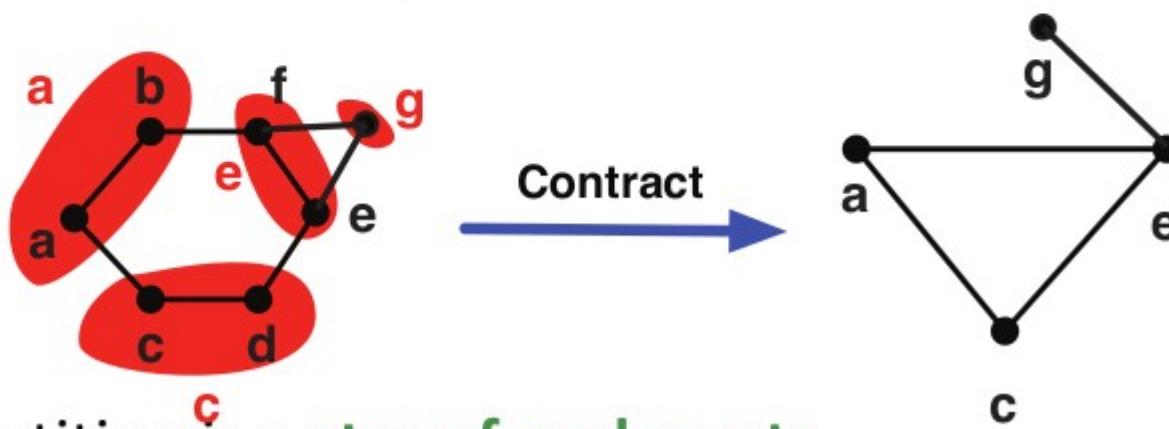


Different Types of Node Partitioning

- Partitions should be **disjoint** and **include all nodes** in G
- Three types of node-partitioning:**
 - Each partition is a **(maximal) clique of nodes**:



- Each partition is a **single node** or **two connected nodes**:

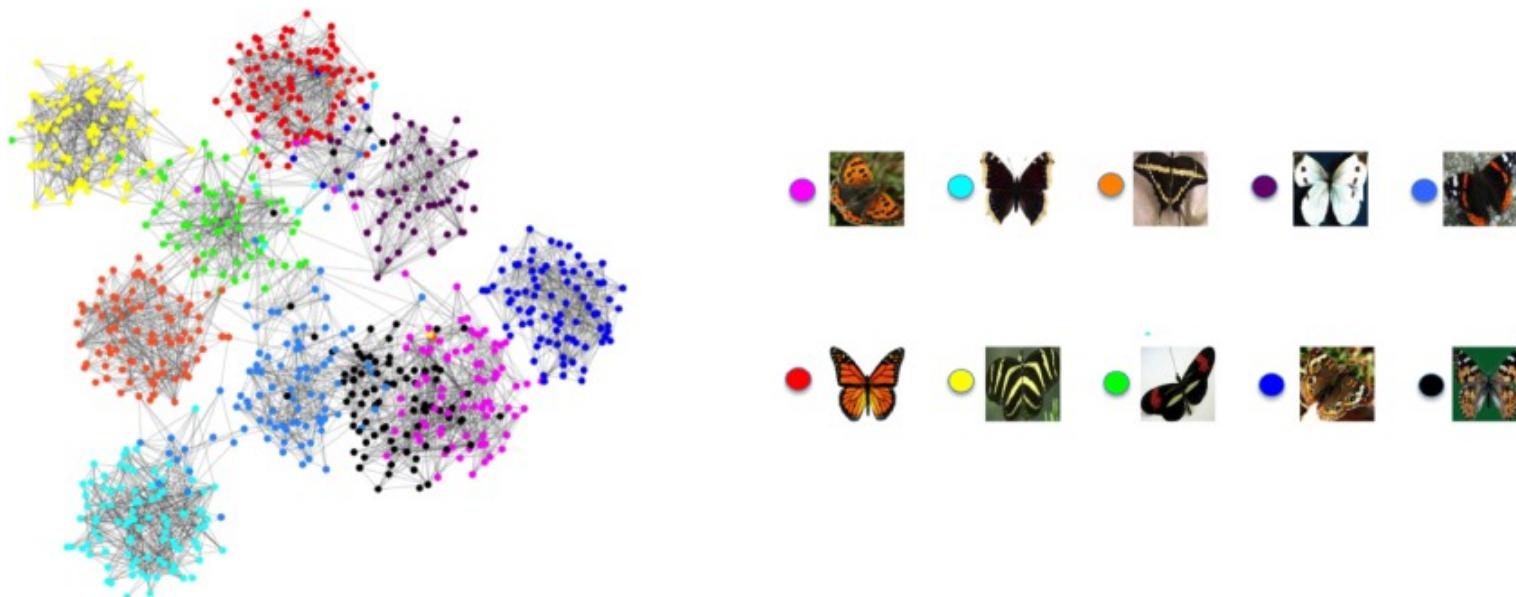


- Each partition is a **star of nodes**, etc.

K-Nearest Neighbors Graph Construction

K-nearest Neighbor Graph

- K-nearest neighbor graph (K-NNG) for a set of objects V is a directed graph with vertex set V :
 - Edges from each $v \in V$ to its K most similar objects in V under a given similarity measure:
 - e.g., Cosine similarity for text
 - e.g., l_2 distance of CNN-derived features for images

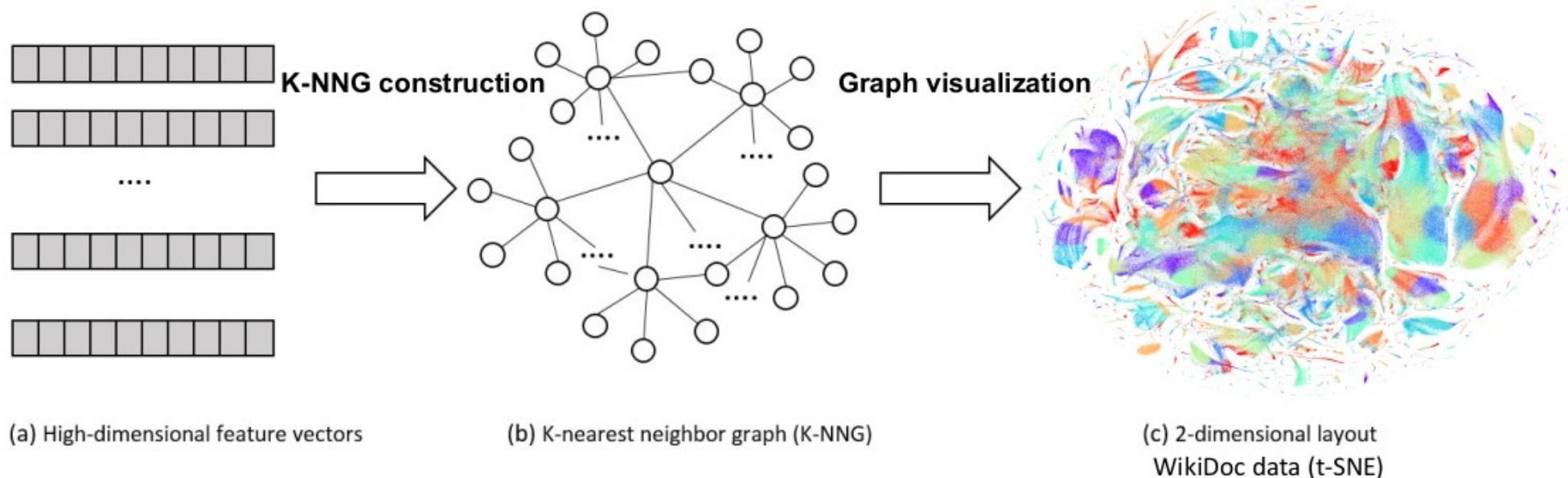


Why should we build K-NNG's?

- K-NNG construction is an **important operation**:
 - **Recommender systems**: connect users with similar product rating patterns, then make recommendations based on the user's graph neighbors
 - **Document retrieval systems**: connect documents with similar content, quickly answer input queries
 - Other problems in **clustering, visualization, information retrieval, data mining, manifold learning**
- K-NNGs allow us to use network methods on datasets with no **explicit graph structure**

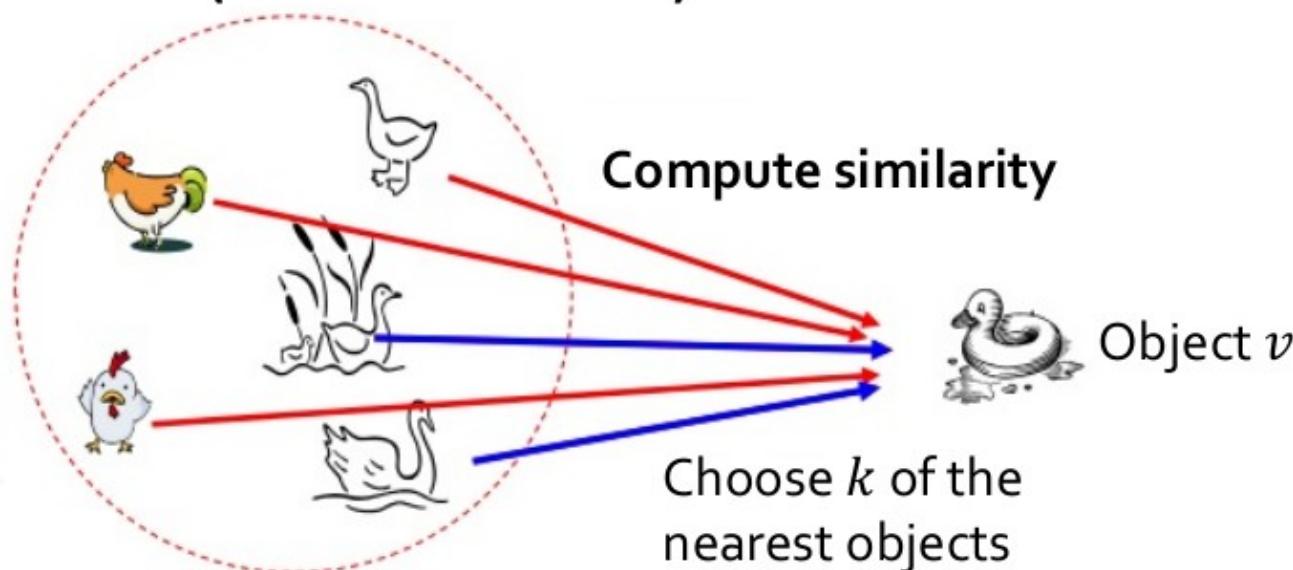
Example: K-NNG in Visualization

- **Problem:** Visualize large high-dim data in 2D space
- **Traditional approach:**
 - Compute similarities between objects
 - Project objects into a 2D space by preserving the similarities
 - **Does not scale** to millions of objects and hundreds of dimensions
- K-NNG can substantially **reduce computational costs**



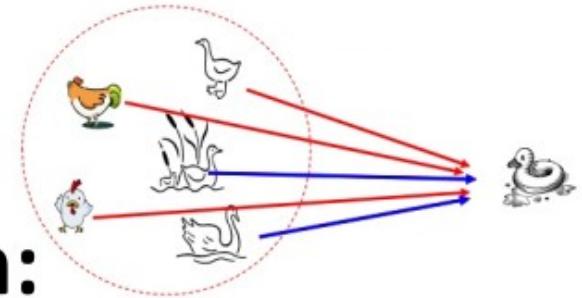
K-NNG: a Brute Force Approach

- Let's construct a K-NNG by **brute-force**:
 - Given n objects V and a distance metric $\sigma: V \times V \rightarrow [0, \infty)$
 - For each possible pair of (u, v) , compute $\sigma(u, v)$
 - For each v , let $B_K(v)$ be v 's K-NN, i.e., the K objects in V (other than v) most similar to v



K-NNG: a Brute Force Approach

- Computational cost of brute-force: $O(n^2)$



- Issues with brute-force approach:

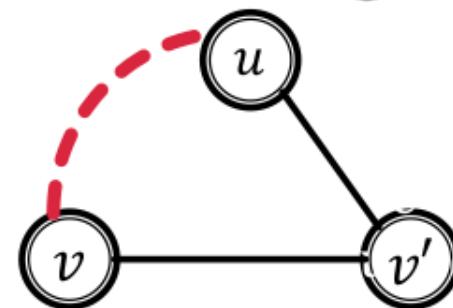
- **Not scalable:** Practical for only small datasets
- **Not general:** Many custom heuristics designed to speed up computations:
 - Many heuristics are specific to a similarity measure
- **Not efficient:** Compute all neighbors for every v
 - We only need k nearest neighbors for every v

Today: NN-Descent Approach

- Can we do better than brute-force?
- Yes, and we will learn about it today!
- **NN-Descent** [Dong et al., WWW 2011]:
 - Efficient algorithm to **approximate K-NNG construction** with **arbitrary similarity measure**
- Other published methods (not covered today):
 - **Locality Sensitive Hashing (LSH)**: A new hash function needs to be designed for a new similarity measure
 - **Recursive Lanczos bisection**: Recursively divide the dataset, so objects in different partitions are not compared
 - **K-NN search problem**: If K-NN problem is solved, K-NNG can be constructed by running a K-NN query for each $v \in V$

NN-Descent: Key Principle

- **Key principle:** A neighbor of a neighbor is also likely to be a neighbor



- Use this principle in a NN-Descent method:
 - Start with an approximation of the K-NNG, B
 - Improve B by **exploring each point's neighbors' neighbors** as defined by the current approximation
 - Stop when no improvement can be made

NN-Descent: Notation

Details

Let:

- V be a metric space with distance metric $d: V \times V \rightarrow [0, \infty)$, $\sigma = -d$ is the similarity measure
- $B_K(v)$ be **v 's K-NN**
- $R_K(v) = \{u \in V; v \in B_K(u)\}$ be **v 's reverse K-NN**
- $B[v]$ be **current approximation** of $B_K(v)$
- $B'[v] = \cup_{v' \in B[v]} B[v']$ be **neighbors of v 's neighbors**
- For any $r > 0$, let **r -ball around v** be:
$$B_r(v) = \{u \in V; d(u, v) \leq r\}$$

NN-Descent: Overview

Details

- **Def:** Metric space V is **growth-restricted** if there exists a constant c , such that:

$$|B_{2r}(v)| \leq c|B_r(v)|, \quad \forall v \in V$$

- The smallest such c is **growing constant** of V

- **Approach:**

- Start with an approximation of the K-NNG, B
 - Use the **growing constant of V** to show that B can be improved by comparing each object v against its current neighbors' neighbors $B'[v]$
- **Next:** Use the **growing-constant argument** on B

NN-Descent: Overview

Details

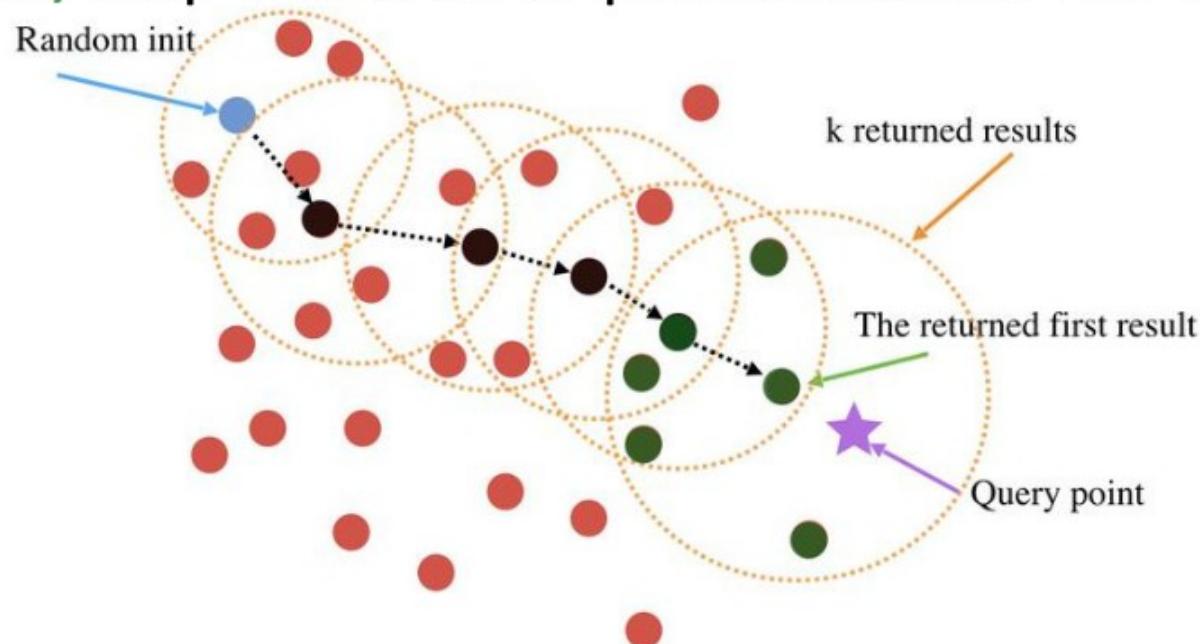
- **Two assumptions:**
 - Let c be the growing constant of V and let $K = c^3$
 - Have an approximate K-NNG B **that is reasonably good:**
 - For a fixed radius r , for all v , $B[v]$ contains K neighbors that are uniformly distributed in $B_r(v)$
- **Lemma:** $B'[v]$ is likely to contain K nearest neighbors in $B_{r/2}(v)$
- **Corollary:** We expect to **halve the maximal distance** to the set of approximate K nearest neighbors by exploring $B'[v]$ for every v

NN-Descent: Algorithm

Details

- **Lemma** suggests the following algorithm:

- Pick a large enough K (depending on **growing constant c**)
- Start from a random K-NNG approximation
- For each v , find K nearest objects by exploring v 's neighbors' neighbors, B'
- **Repeat;** stop when no improvement can be made



NN-Descent: Algorithm

Details

Algorithm 1: NNDESCENT

Data: dataset V , similarity oracle σ , K

Result: K-NN list B

begin

$B[v] \leftarrow \text{SAMPLE}(V, K) \times \{\infty\}, \quad \forall v \in V$

loop

$R \leftarrow \text{REVERSE}(B)$

$\bar{B}[v] \leftarrow B[v] \cup R[v], \quad \forall v \in V;$

$c \leftarrow 0 \quad //\text{update counter}$

for $v \in V$ **do**

for $u_1 \in \bar{B}[v], u_2 \in \bar{B}[u_1]$ **do**

$l \leftarrow \sigma(v, u_2)$

$c \leftarrow c + \text{UPDATENN}(B[v], \langle u_2, l \rangle)$

return B **if** $c = 0$



A. Start by picking a random approximation of K-NN for each object



B. Improve the approximation by comparing each object against its current neighbors' neighbors, including K-NN and reverse K-NN



C. Stop when no improvement can be made

function SAMPLE(S, n)

return Sample n items from set S

function REVERSE(B)

begin

$R[v] \leftarrow \{u \mid \langle v, \dots \rangle \in B[u]\} \quad \forall v \in V$

return R

function UPDATENN($H, \langle u, l, \dots \rangle$)

 Update K-NN heap H ; return 1 if changed, or 0 if not.

Experimental Setup: Data

■ Datasets:

- **Corel**: Each **image** is segmented into 14 regions, a feature is extracted from each region
- **Audio**: Each **sentence** is described by 192 features
- **Shape**: Each **shape** is described by 544-dim feature vector
- **DBLP**: Each **record** includes authors' names and pub. title
- **Flickr**: Each **image** is segmented into regions, a pixel-based feature is extracted from each region

■ Similarity measures: L1, L2, Cosine, Jaccard, EMD

Dataset	# Objects	Dimension	Similarity Measures
Corel	662,317	14	l_1, l_2
Audio	54,387	192	l_1, l_2
Shape	28,775	544	l_1, l_2
DBLP	857,820	N/A	Cosine, Jaccard
Flickr	100,000	N/A	EMD

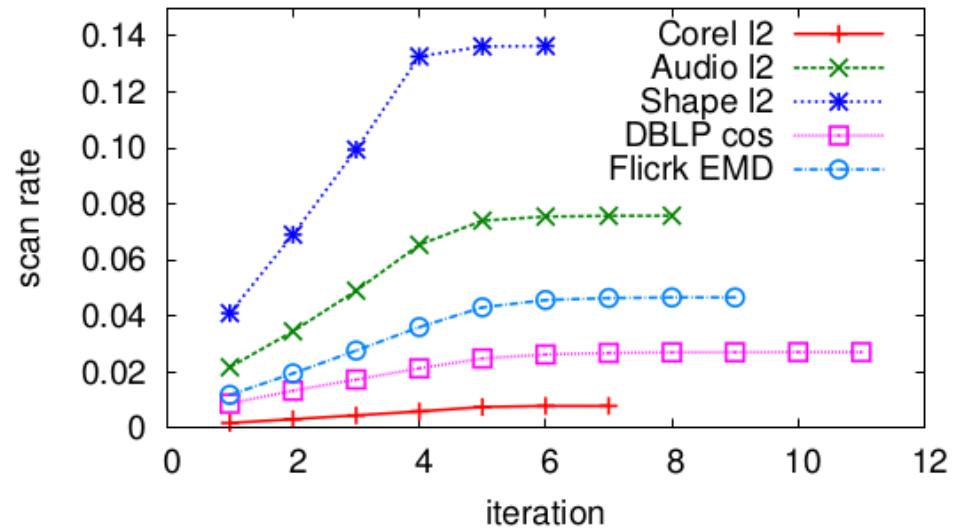
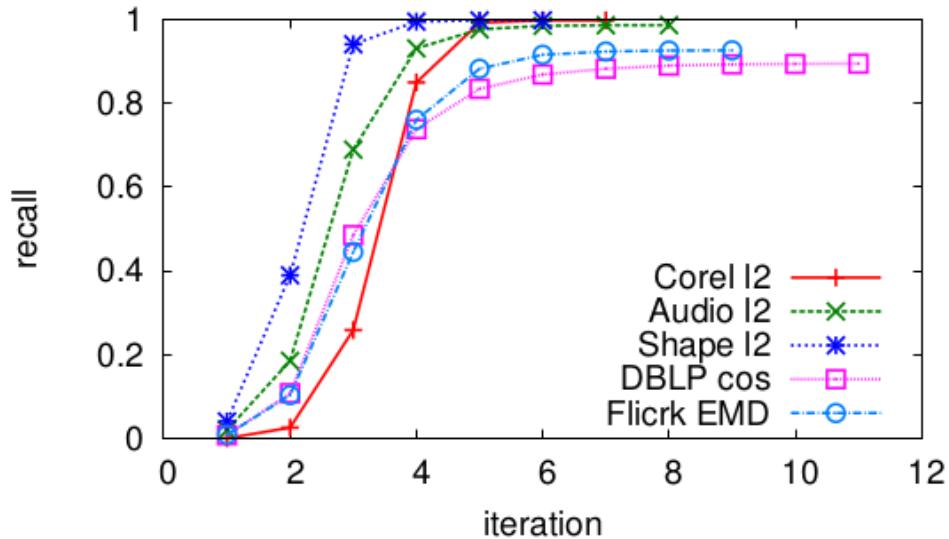
(EMD: earth mover's distance)

Experimental Setup: Measures

- Use recall as an **accuracy measure**:
 - **Ground-truth**: true K-NNs obtained by scanning the datasets in brute force
 - **Recall of one object** is the number of its true K-NN members found divided by K
 - **Recall of an approximate K-NNG** is the average recall of all objects
- Use #(sim. evaluations) as a **measure of computational cost**:

$$\text{scan rate} = \frac{\text{\#(similarity evaluations)}}{n(n - 1)/2}$$

Exp: Overall Performance

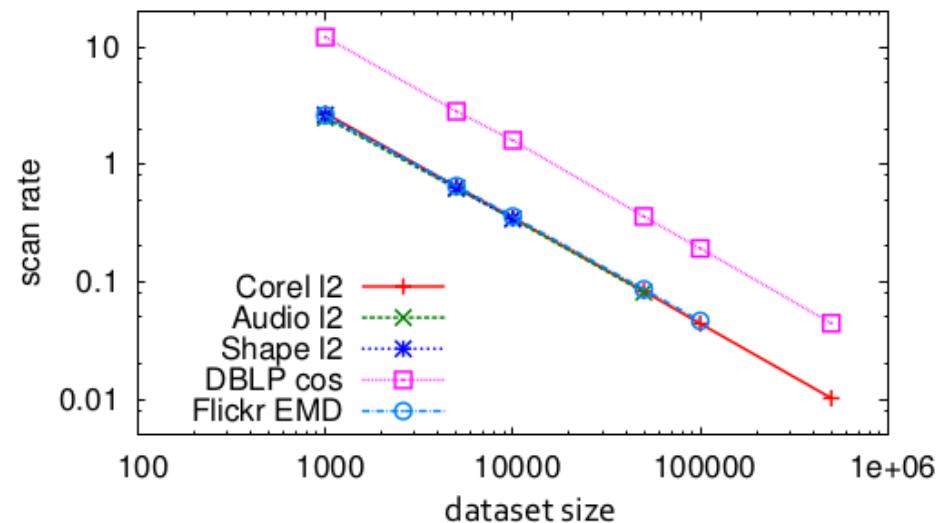


- Similar performance trends on different datasets
- Fast convergence across all datasets:
 - Curves are close to their final recall after 5 iterations
 - All curves converge within 12 iterations

Exp: Performance as Data scales

Size	Corel l_2	Audio l_2	Shape l_2	DBLP cos	Flickr EMD
1K	1.000	0.999	1.000	0.959	0.999
5K	1.000	0.996	0.992	0.970	0.991
10K	1.000	0.993	0.998	0.970	0.983
50K	0.999	0.988	-	0.951	0.953
100K	0.999	-	-	0.940	0.925
500K	0.997	-	-	0.907	-

(recall values)

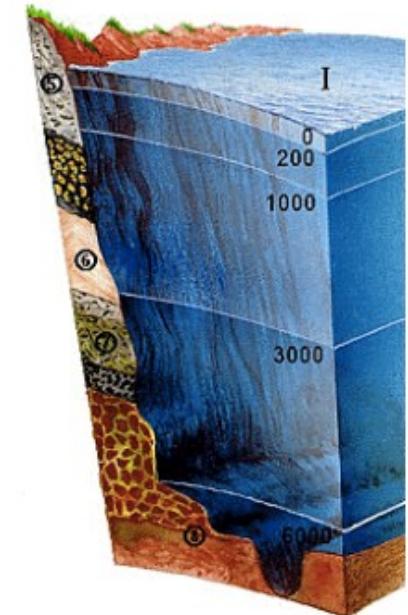


- Run experiments on samples of the full datasets and observe changes in recall and scan rate as sample size grows
- Results:**
 - As dataset grows, there is only a minor decline in recall
 - All curves form parallel straight lines in the scan rate vs. dataset size:
 - NN-descent has a polynomial time complexity
 - Fit the scan rate curves to obtain empirical complexity of NN-Descent:
 - $O(n^{1.14}) \ll O(n^2)$ (=brute-force)

Network Deconvolution

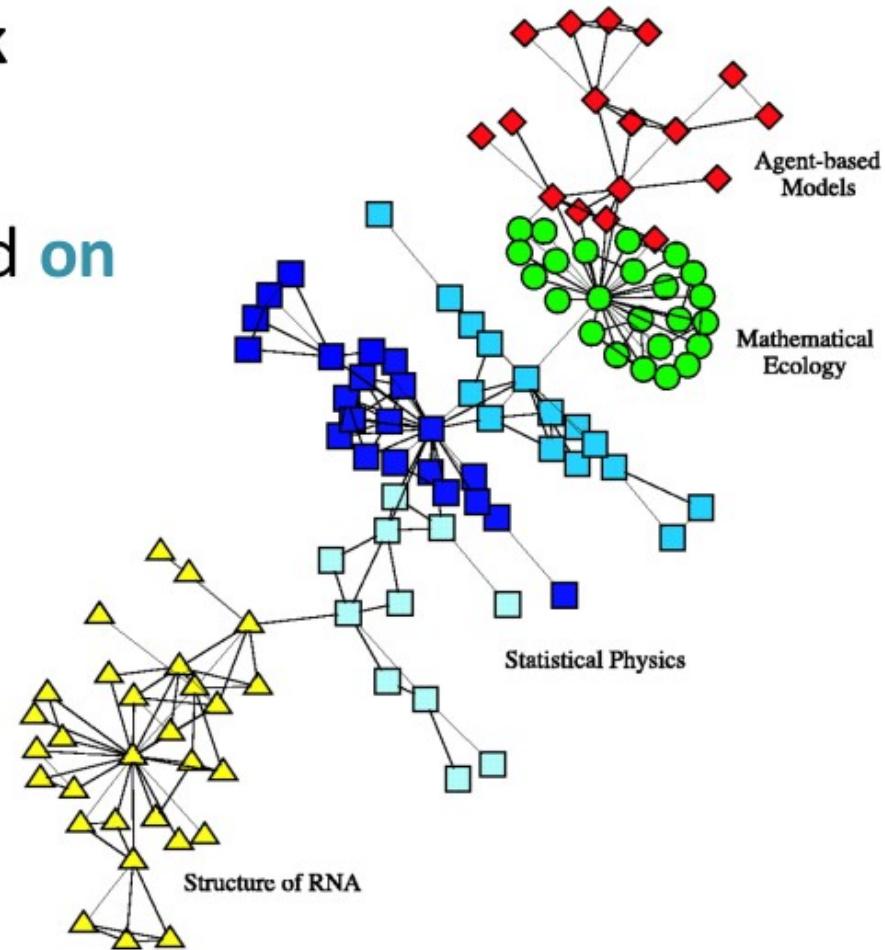
Motivation

- Networks represent **dependencies among objects**:
 - Co-authorships between scientists
 - Friendships between people
 - Who-eats-whom in food webs
 - Bonds between molecular residues
 - Regulatory relationships between genes
- **Indirect dependencies** occur because of **transitive effects of correlation**
- **Problem:** How to separate direct dependencies from indirect ones?



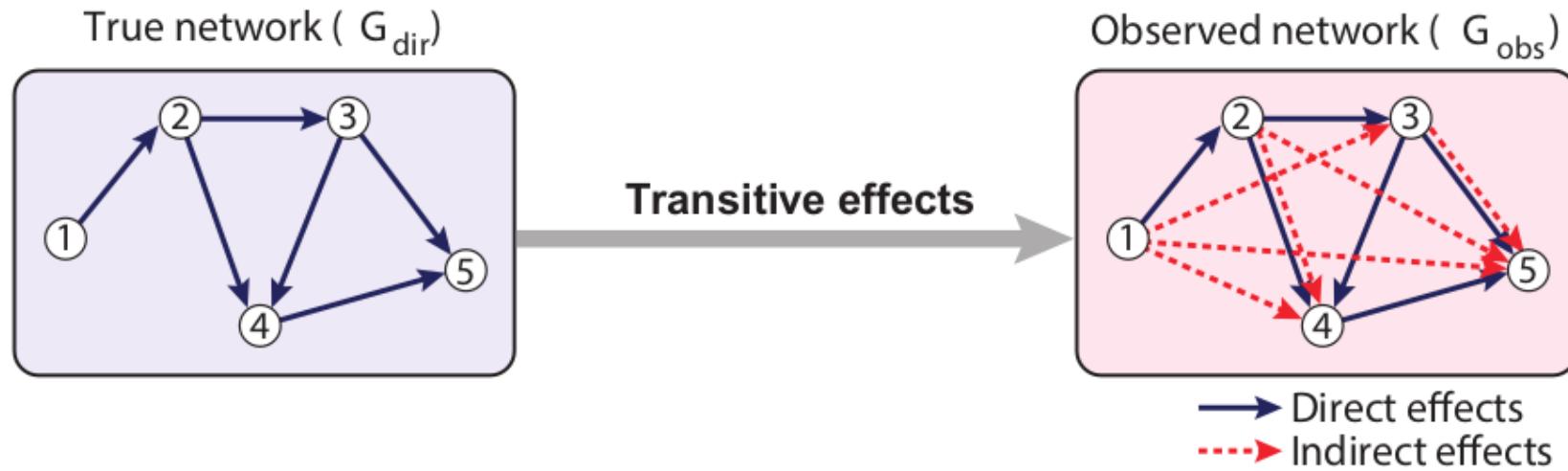
Application: Co-Authorship

- **Goal:** Distinguish **strong** and **weak** collaborations between scientists
- Collaboration tie strengths depend **on publication details**, such as:
 - #(papers) each pair of scientists has collaborated on
 - #(co-authors) on each of the papers
- Strength of ties are important for:
 - **Recommending friends** and colleagues
 - Recognizing **conflicts of interest**
 - Evaluating authors' **contribution to teams**



Observed Network

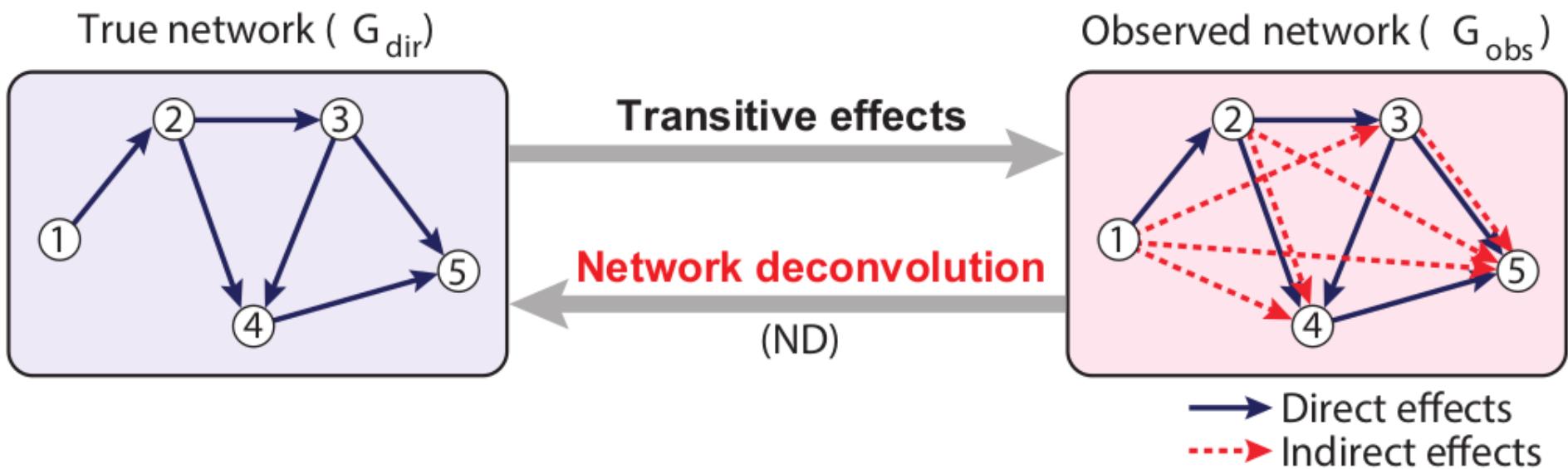
- **Observed network:** Combined direct and indirect effects:



- **Indirect edges** might be due to higher-order interactions (e.g., $1 \rightarrow 4$)
- Each edge might contain **both direct and indirect components** (e.g., $2 \rightarrow 4$)

Network Deconvolution

- **Goal:** Reverse the effect of transitive information flow across all indirect paths:
 - Recover **true direct network** (**blue edges**, G_{dir}) based on **observed network** (**combined blue and red edges**, G_{obs})

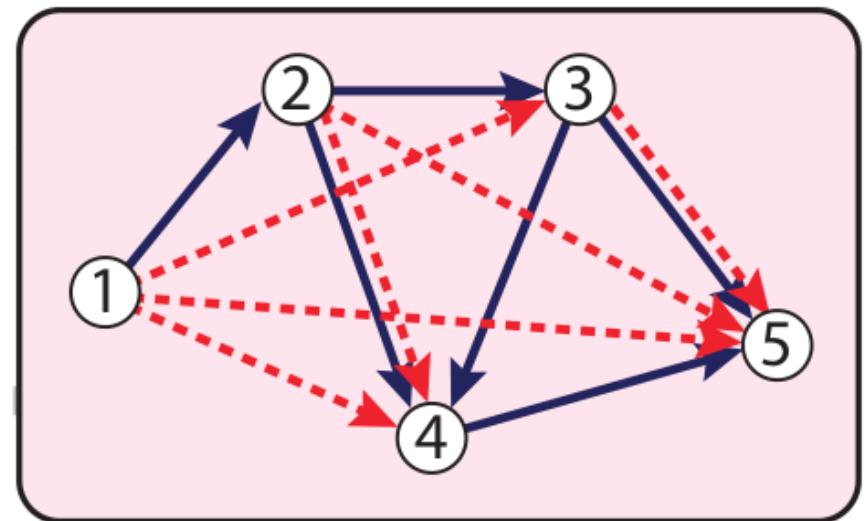


Feizi *et al.*, *Nature Biotechnology*, 31:8, 2013.

Network Deconvolution: Challenge

- Direct edges in a network can lead to indirect relationships:
 - **Transitive information flow**
- **Indirect effects** can be of **length**:
 - 2 (e.g., $1 \rightarrow 2 \rightarrow 3$)
 - 3 (e.g., $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$)
 - **higher-order**
- Indirect effects can combine:
 - **Both direct and indirect** effects (e.g., $2 \rightarrow 4$)
 - Multiple indirect effects along **varying paths** (e.g., $2 \rightarrow 3 \rightarrow 5$, $2 \rightarrow 4 \rightarrow 5$)

Observed network (G_{obs})



→ Direct effects
→ Indirect effects

Net. Deconvolution: Formally

Details

- Transitive effects in G_{obs} can be expressed as an infinite sum of G_{dir} and all indirect effects:

$$G_{\text{obs}} = G_{\text{dir}} + G_{\text{indir}}$$

- Indirect effects can be of **increasing lengths**:

$$G_{\text{indir}} = G_{\text{dir}}^2 + G_{\text{dir}}^3 + G_{\text{dir}}^4 + \dots$$

\downarrow \downarrow \downarrow
2nd order 3rd order 4th order

- 2nd order effects:** $G_{\text{dir}}^2 = A_{\text{dir}}^2$

- The number of edges in G_{obs} of indirect paths of length 2**

- 3rd order effects:** $G_{\text{dir}}^3 = A_{\text{dir}}^3$

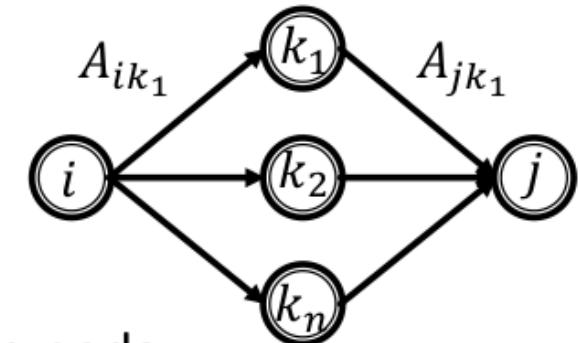
- The number of edges in G_{obs} of indirect paths of length 3**

Powers of Adjacency Matrices

- Let's raise adjacency matrix A_{dir} to the second power:

- The (i, j) -th entry of A_{dir}^2 is:

$$A_{\text{dir}}^2(i, j) = \sum_{k=1}^n A_{\text{dir}}(i, k) A_{\text{dir}}(k, j)$$



- This sum is only greater than zero if there exists a node k for which $A_{\text{dir}}(i, k)$ and $A_{\text{dir}}(k, j)$ are both nonzero:
 - There exists a node k that is connected to both nodes i and j
 - The sum counts the number of neighbors that nodes i and j share
 - The sum counts the paths of length 2 between nodes i and j

- This reasoning is valid for higher powers of A_{dir} :

- $A_{\text{dir}}^3(i, j)$ counts the paths of length 3 between i and j
- $A_{\text{dir}}^4(i, j)$ counts the paths of length 4 between i and j

Net. Deconvolution: Formally

Details

- Idea: Model indirect flow as **power series** of direct flow:

$$G_{\text{obs}} = G_{\text{dir}} + G_{\text{dir}}^2 + G_{\text{dir}}^3 + G_{\text{dir}}^4 + \dots$$



Transitive closure of G_{dir}

- Note: Linear scaling of G_{obs} so that max absolute eigenvalue of $G_{\text{dir}} < 1$:

- Indirect effects decay exponentially with path length
- Infinite series converges

Net. Deconvolution: Formally

Details

- **Transitive closure of G_{dir}** can be expressed as an infinite sum of:
 - True direct network, G_{dir}
 - All indirect effects along paths of increasing lengths, $G_{\text{dir}}^2, G_{\text{dir}}^3, G_{\text{dir}}^4, \dots$
- **Idea:** Can be written in a closed form as an infinite-series sum using **Taylor series expansions**:

$$G_{\text{obs}} = G_{\text{dir}} + G_{\text{dir}}^2 + G_{\text{dir}}^3 + G_{\text{dir}}^4 + \dots = \\ G_{\text{dir}}(I + G_{\text{dir}} + G_{\text{dir}}^2 + G_{\text{dir}}^3 + \dots) = G_{\text{dir}}(I - G_{\text{dir}})^{-1}$$

Note: Let X be any square matrix with max absolute eigenvalue < 1 . Then the following series converges: $I + X + X^2 + X^3 + \dots$
The series converges to: $\sum_{k=0}^{\infty} X^k = (1 - X)^{-1}$

Net. Deconvolution: Formally

Details

- Using **Taylor series expansions** we get a **closed-form expression for G_{obs}** :

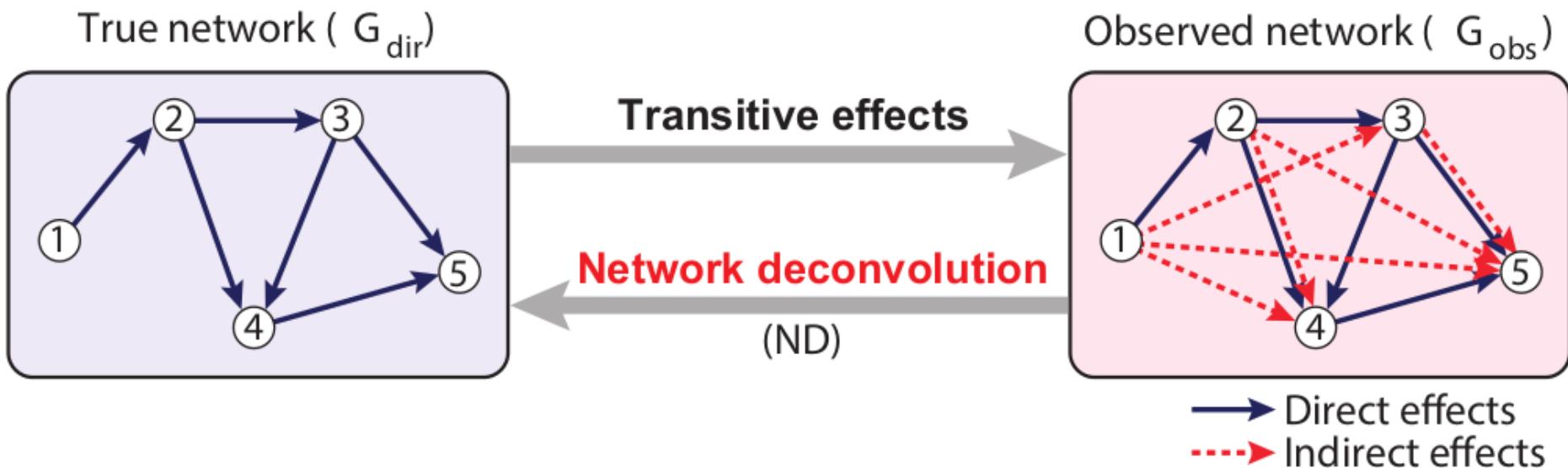
$$G_{\text{obs}} = G_{\text{dir}}(I - G_{\text{dir}})^{-1}$$

- **In network deconvolution:**
 - Observed network G_{obs} is known
 - True direct network G_{dir} needs to be **recovered**
- Finally, we get a **closed-form solution for G_{dir}** :

$$G_{\text{dir}} = G_{\text{obs}}(I + G_{\text{obs}})^{-1}$$

Net. Deconvolution: Recap

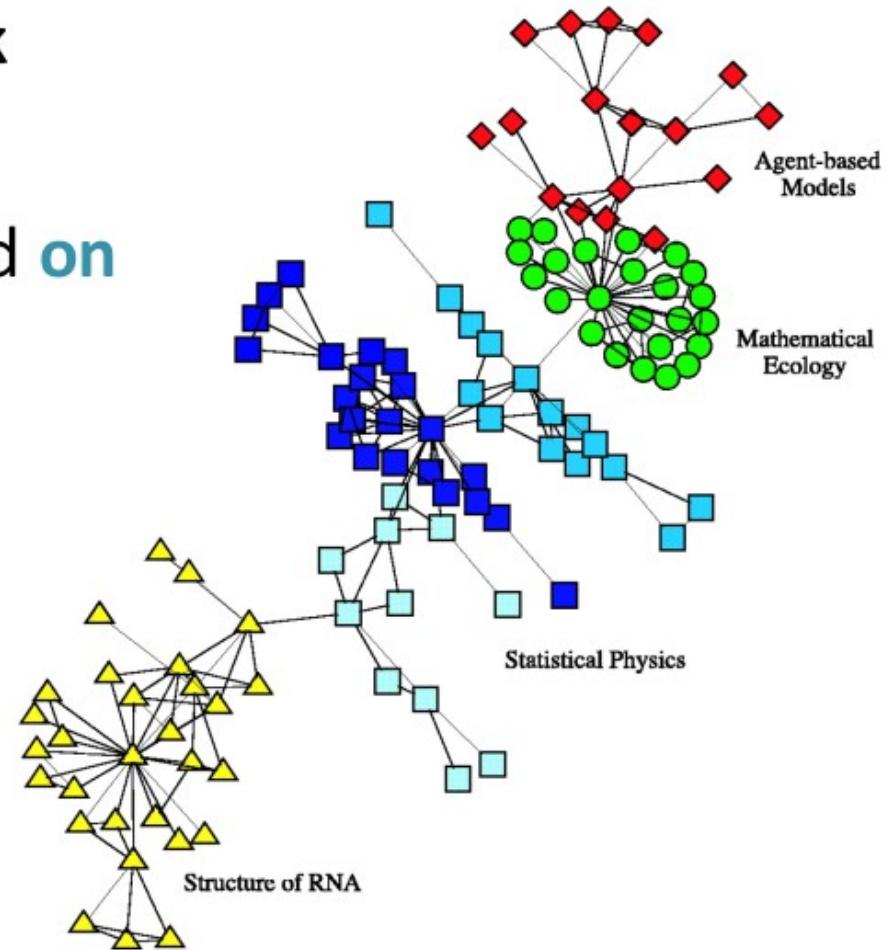
- Use **closed-form expression for G_{obs}** to **recover** true direct network G_{dir}



Transitive closure:	$G_{\text{obs}} = G_{\text{dir}} + \overbrace{G_{\text{dir}}^2 + G_{\text{dir}}^3 + \dots}^{\text{Indirect effects}} = G_{\text{dir}}(I - G_{\text{dir}})^{-1}$
Network deconvolution:	$G_{\text{dir}} = G_{\text{obs}}(I + G_{\text{obs}})^{-1}$

Application: Co-Authorship

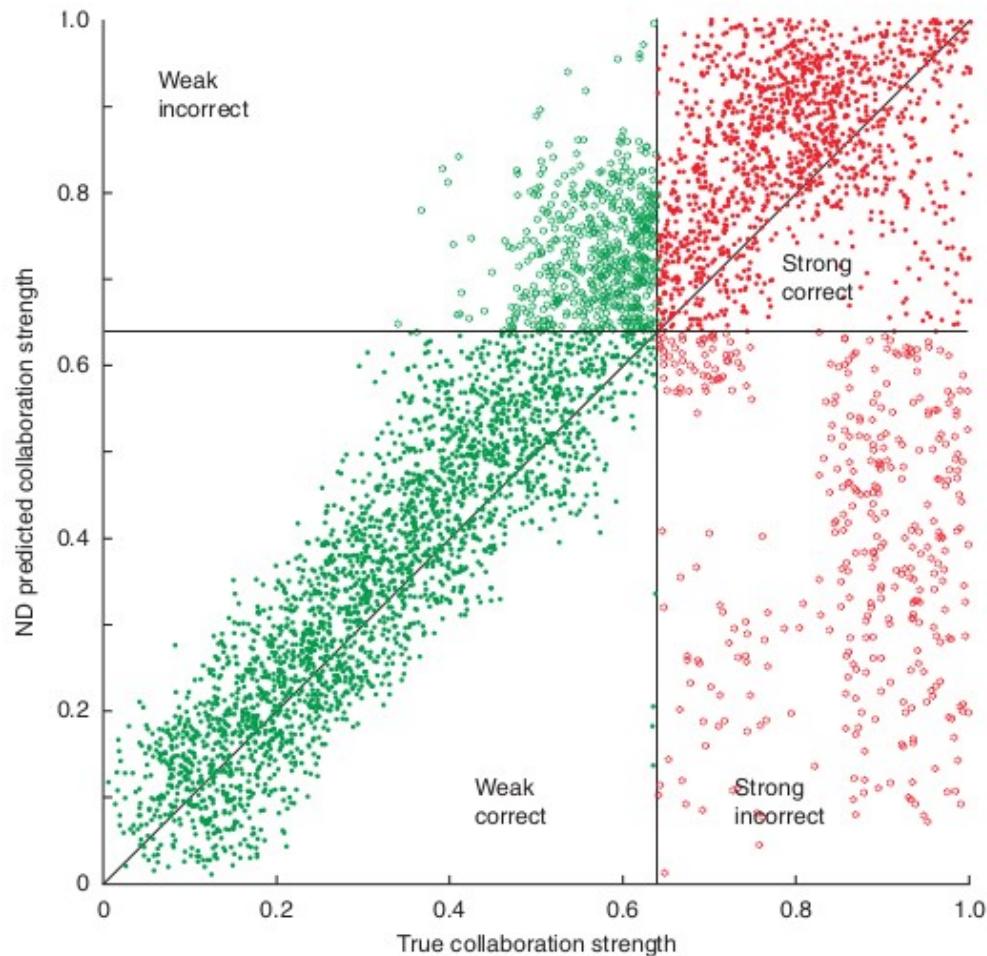
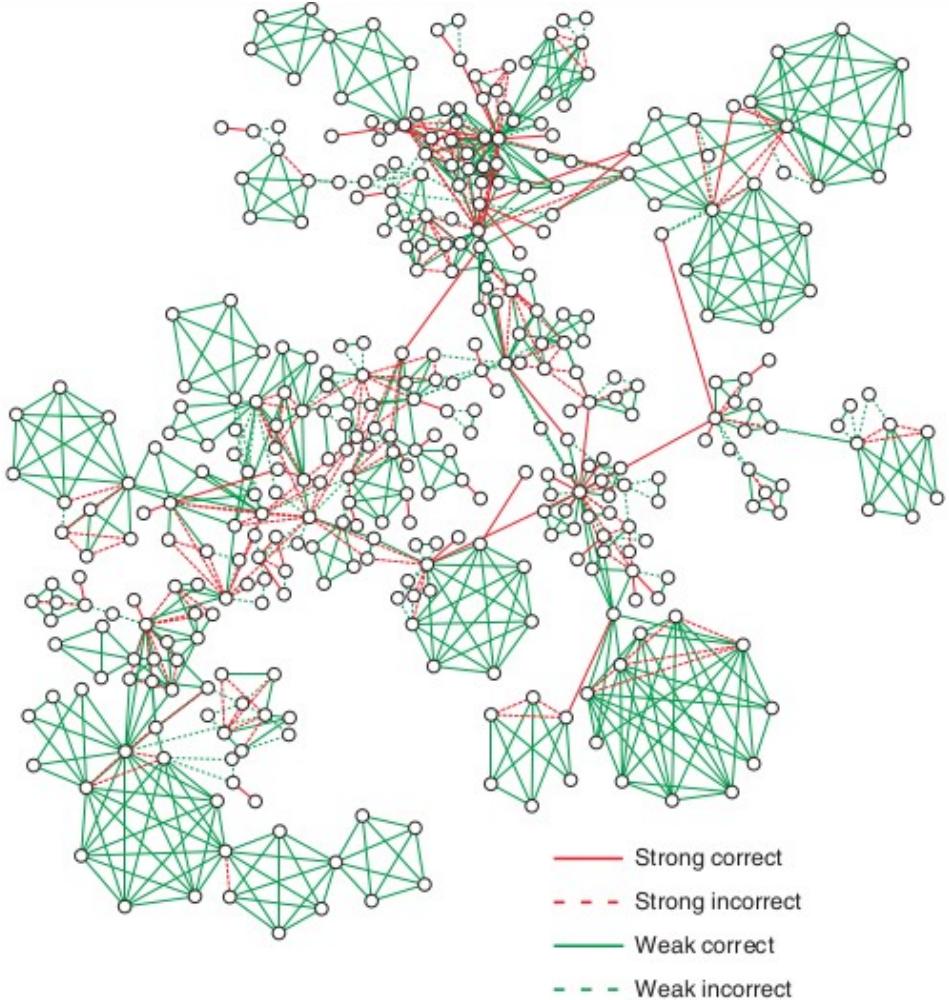
- **Goal:** Distinguish **strong** and **weak** collaborations between scientists
- Collaboration tie strengths depend **on publication details**, such as:
 - #(papers) each pair of scientists has collaborated on
 - #(co-authors) on each of the papers
- Strength of ties are important for:
 - **Recommending friends** and colleagues
 - Recognizing **conflicts of interest**
 - Evaluating authors' **contribution to teams**



Application: Co-Authorship

- **Data:** Unweighted network of **scientists** working in the field of network science:
 - Two authors are linked if they co-authored at least one paper
- **Setup:** Apply ND on the co-authorship network:
 - ND returns a weighted network whose:
 - Transitive closure most closely captures the input network
 - Weights represent the inferred strength of direct interactions
 - **Output:** Rank co-authorship edges by **the ND-assigned weights**
- **Ground-truth data:**
 - **True collaboration strengths** are computed by summing the number of co-authored papers and down-weighting each paper by the number of additional co-authors
 - Compute **correlation** between **ND-assigned weights** and **true collaboration strengths**

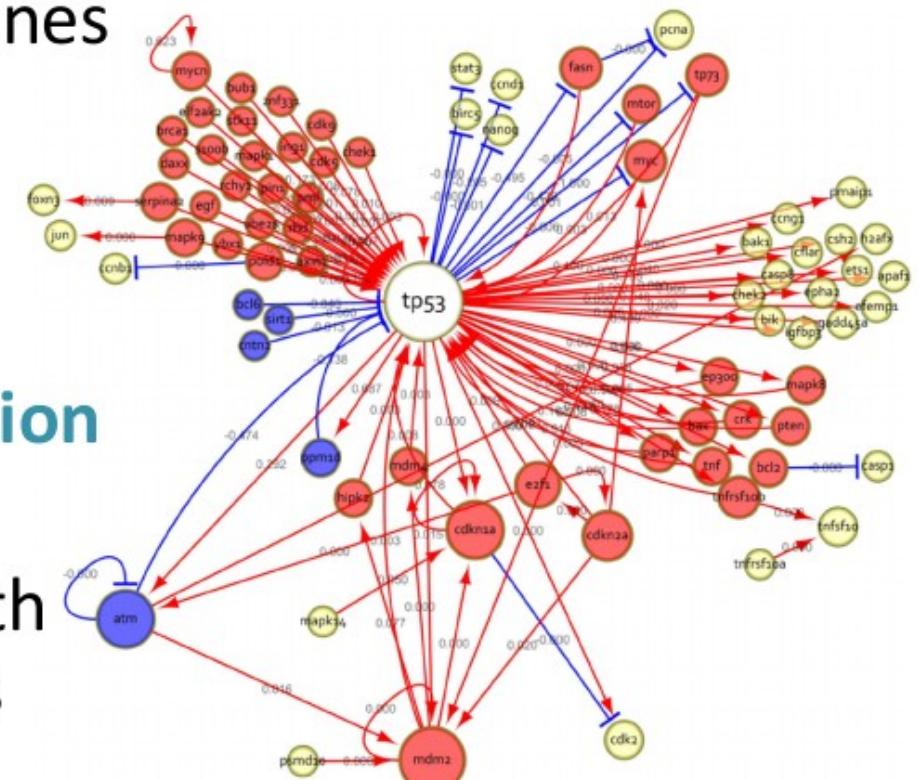
Co-Authorship: Results



- Agreement between the rank obtained by the true collaboration strength and the rank provided by the ND weight, $R^2 = 0.76$
- **Conclusion:** ND predict collaboration tie strengths **solely by using network topology** (i.e., not using other publication details)

Application: Gene Network

- **Goal:** Infer a **gene regulatory network** from gene feature vectors describing gene activity:
 - **Nodes** represent genes
 - **Edges** represent regulatory relationships between regulators and their target genes
- Well-studied **problem in bioinformatics**:
 - A dataset is a **gene-by-condition expression matrix**
 - Expression matrix is **noisy** with many **indirect** measurements

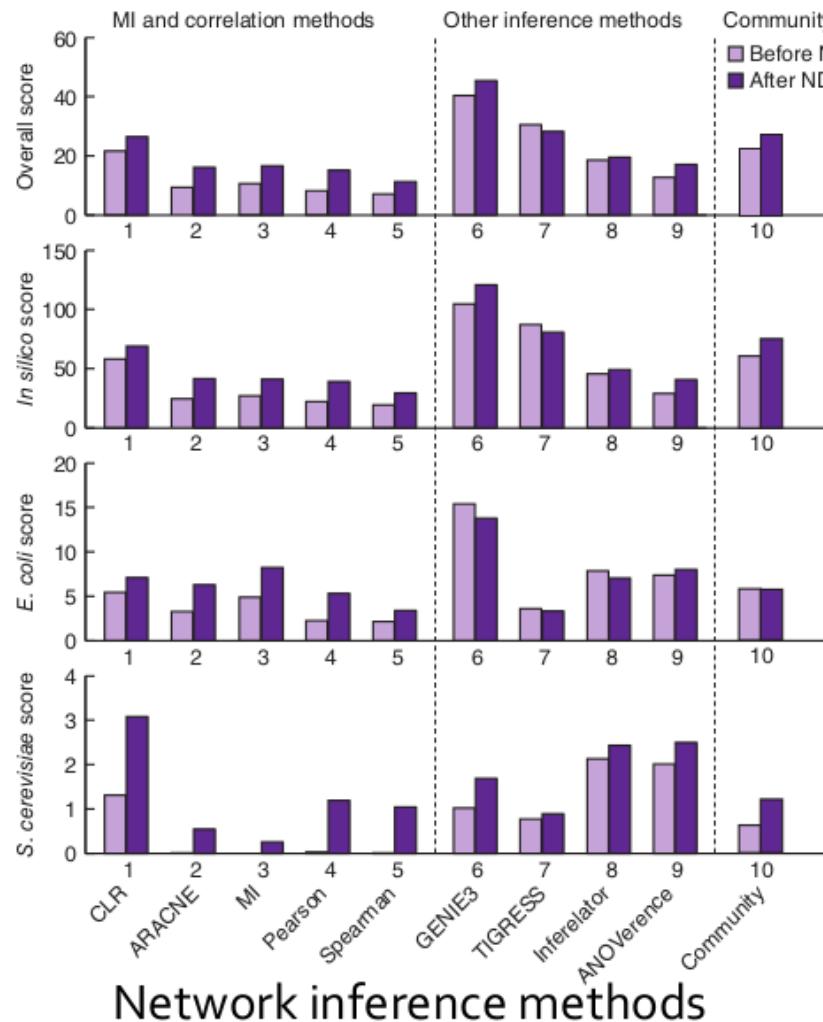


Gene Network: Data

- **3 datasets:** Gene expression datasets from: bacterium *E. coli*, yeast *S. cerevisiae*, and a simulated env (*in silico*)
- **Setup:** Use ND to improve network inference methods by eliminating indirect edges in the inferred networks:
 1. Infer a gene regulatory network using a particular network inference method
 2. Apply ND to the inferred network to deconvolve the network
 3. Evaluate deconvolved network against ground-truth data
- **Ground-truth data:**
 - True positive regulatory relationships (*i.e.*, edges) are defined as a set of interactions experimentally validation in a laboratory

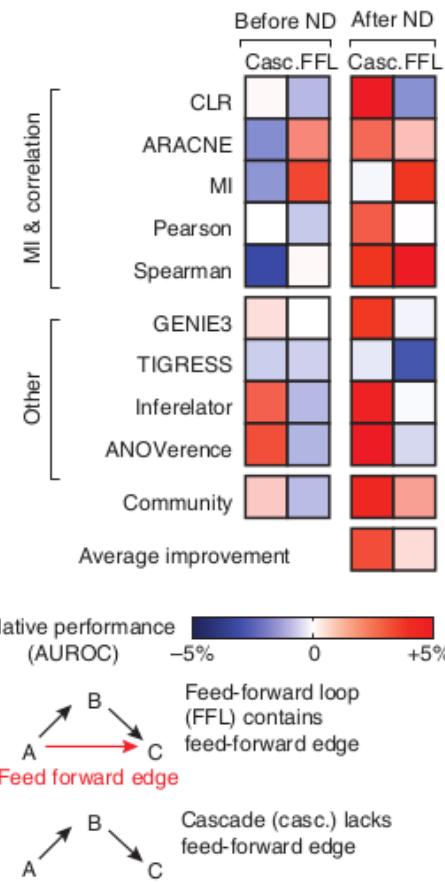
Gene Network: Results

Datasets



Network inference methods

ND improves the performance of top-performing network inference methods



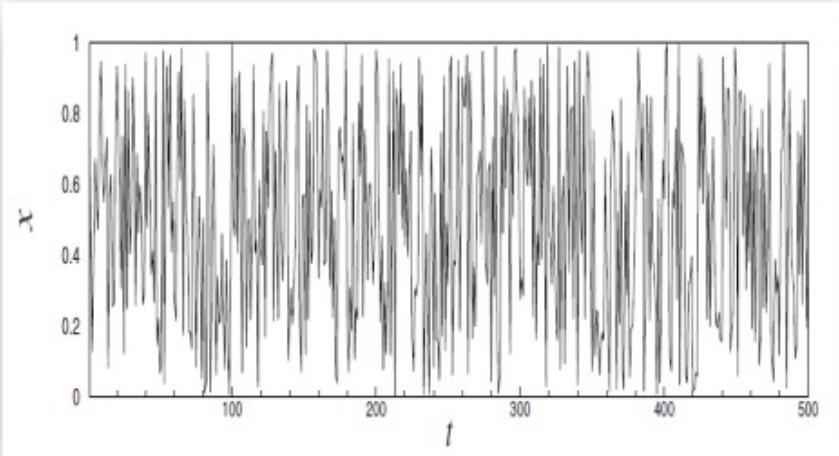
Relative performance of inference methods for cascades (**casc.**) and feed-forward loops (**FFL**) before and after network deconvolution

Network Deconvolution: Recap

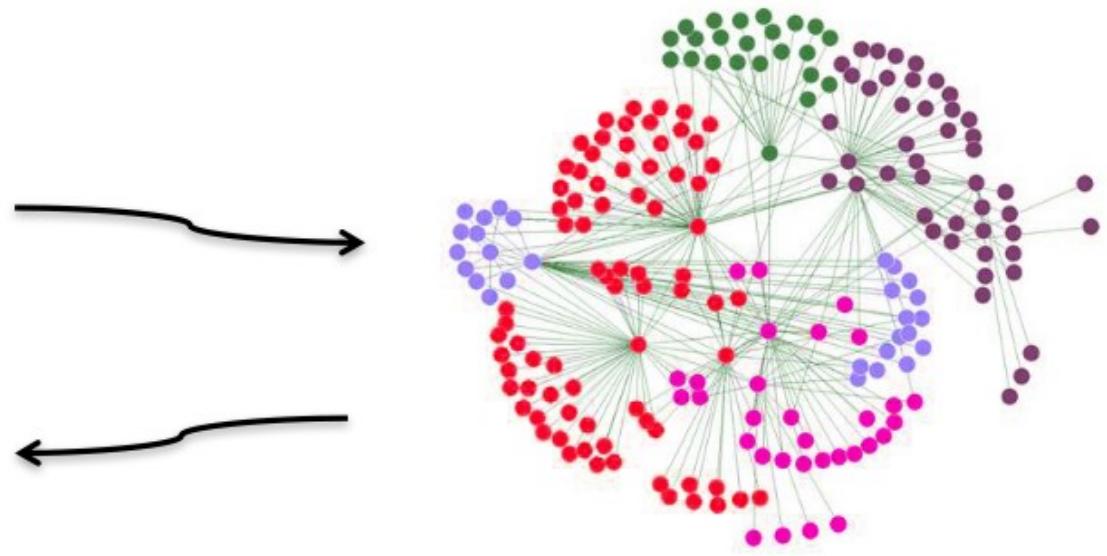
- General approach to **identify direct dependencies between objects in a network**:
 - Remove **spurious edges** that are due to indirect effects
 - Decrease **over-estimated edge weights**
 - **Rescale edge weights** so that they correspond to direct dependencies between objects
- Other published methods (not covered today):
 - Partial correlations and random matrix theory
 - Graphical models, *e.g.*, Graphical lasso, Bayesian nets, Markov random fields
 - Causal inference models

Time Series meets Network Science

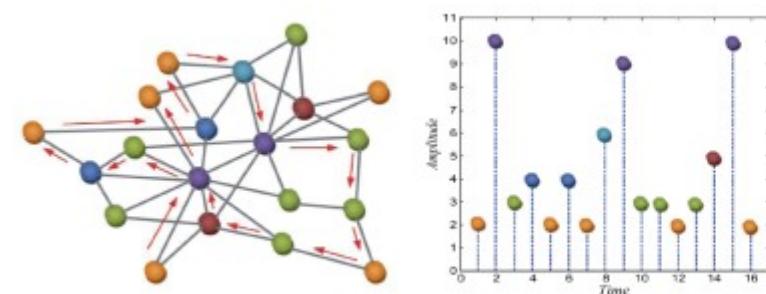
Time Series and Network Science



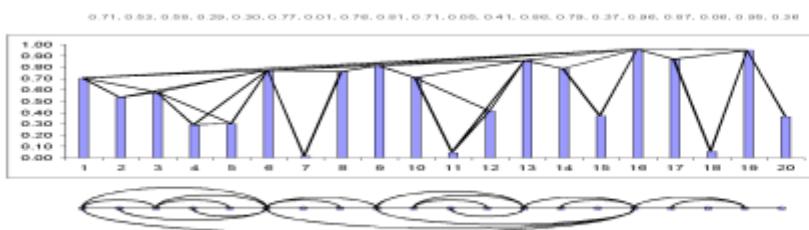
SIGNAL PROCESSING



NETWORK SCIENCE



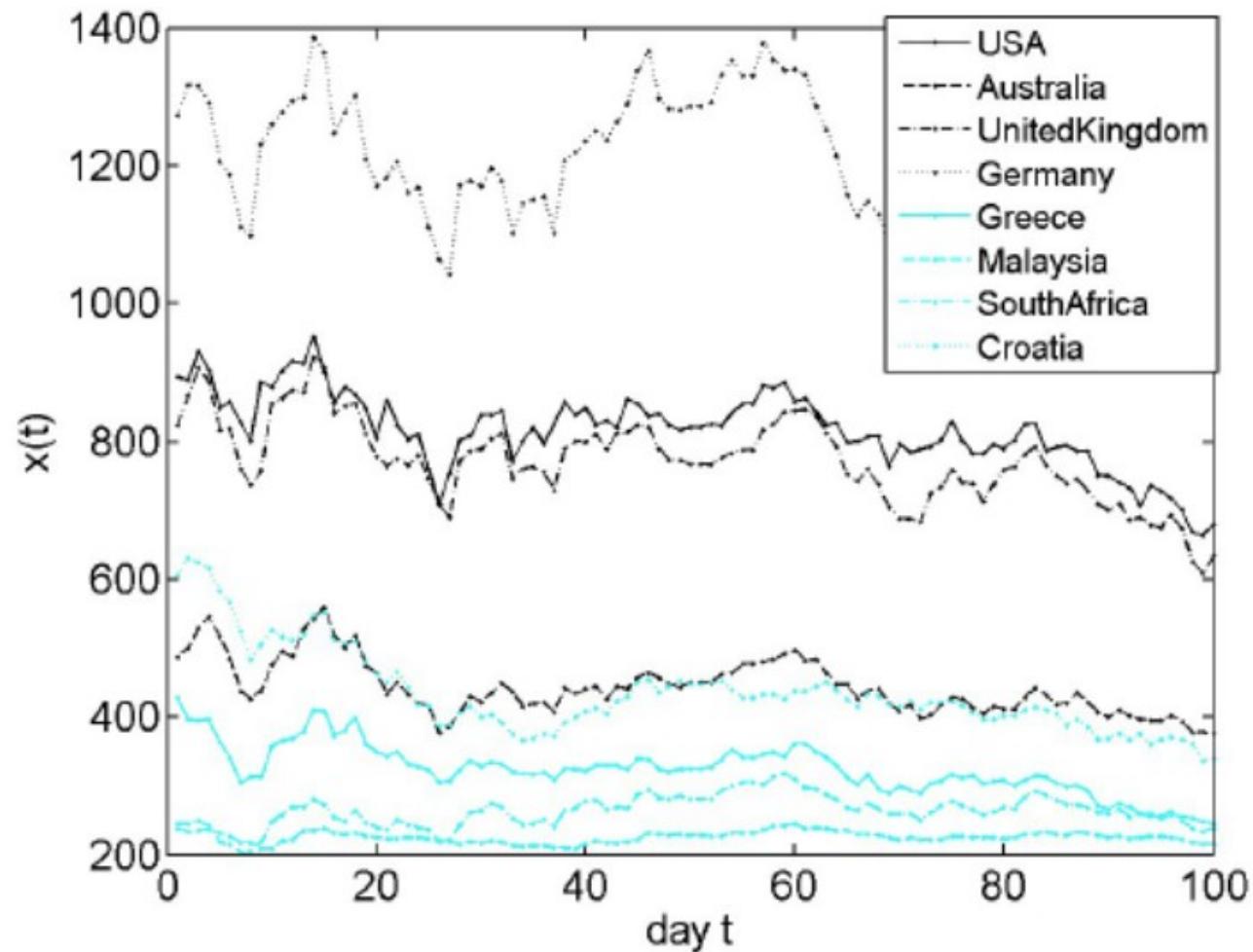
Signal processing of graphs



Graph-theoretical time series analysis

Correlation and Functional Networks

$N = 8$ world stock markets, daily indices, $n = 100$ days.



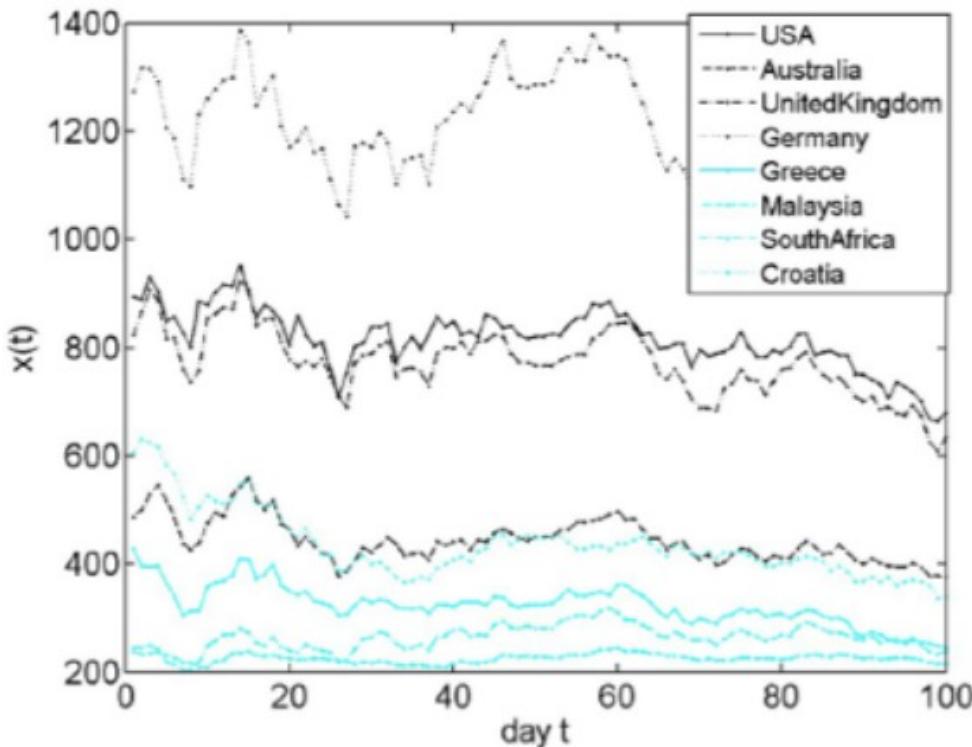
Similar indices, links among world stock markets?

Correlation and Functional Networks

A similarity measure $\text{sim}(i, j)$ quantifies the level of

- correlation or coupling between X_i and X_j (undirected link)
- causality from X_i and X_j , and vice versa (directed link).

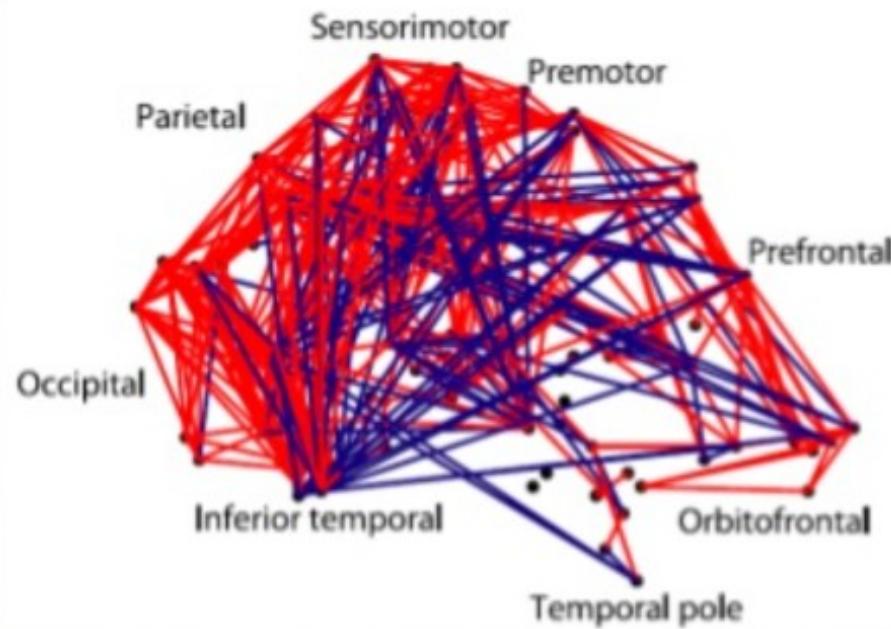
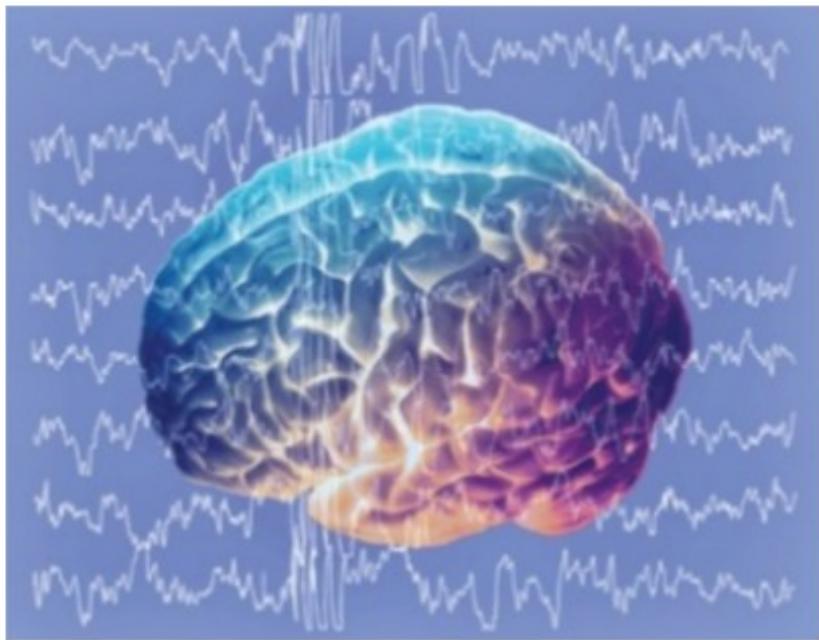
A standard similarity measure is again $\text{Corr}(X_i, X_j) = r_{X_i, Y_j}$.



One can interpret this matrix as
a weighted adjacency matrix!

Correlation network

Correlation and Functional Networks

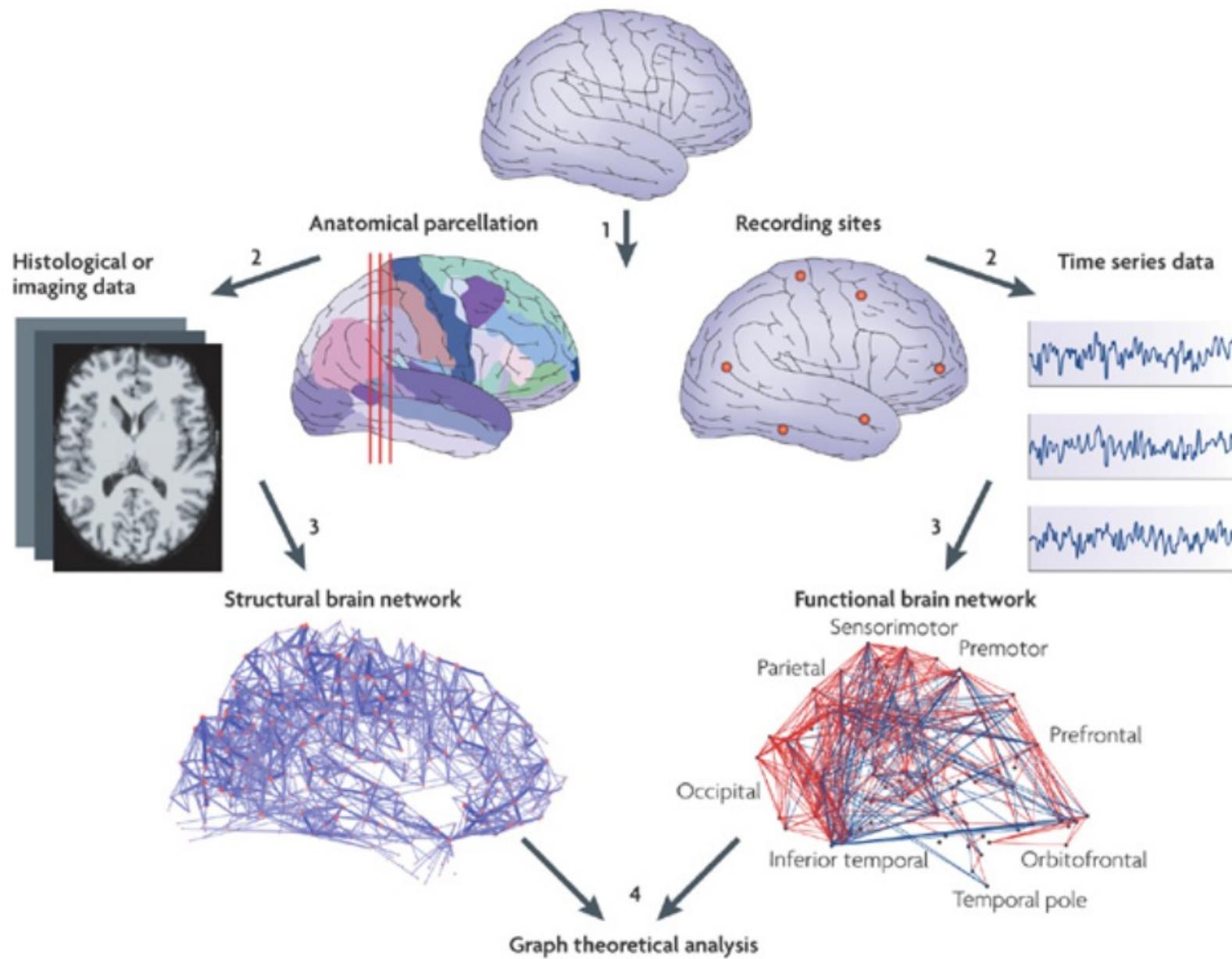


Functional networks

One can measure signals from the brain (EEG, fmri) at different regions and extract a correlation network from the multivariate time series.

This network describes correlations between the activity of different regions of the brain, and it's called a **functional network**.

Correlation and Functional Networks

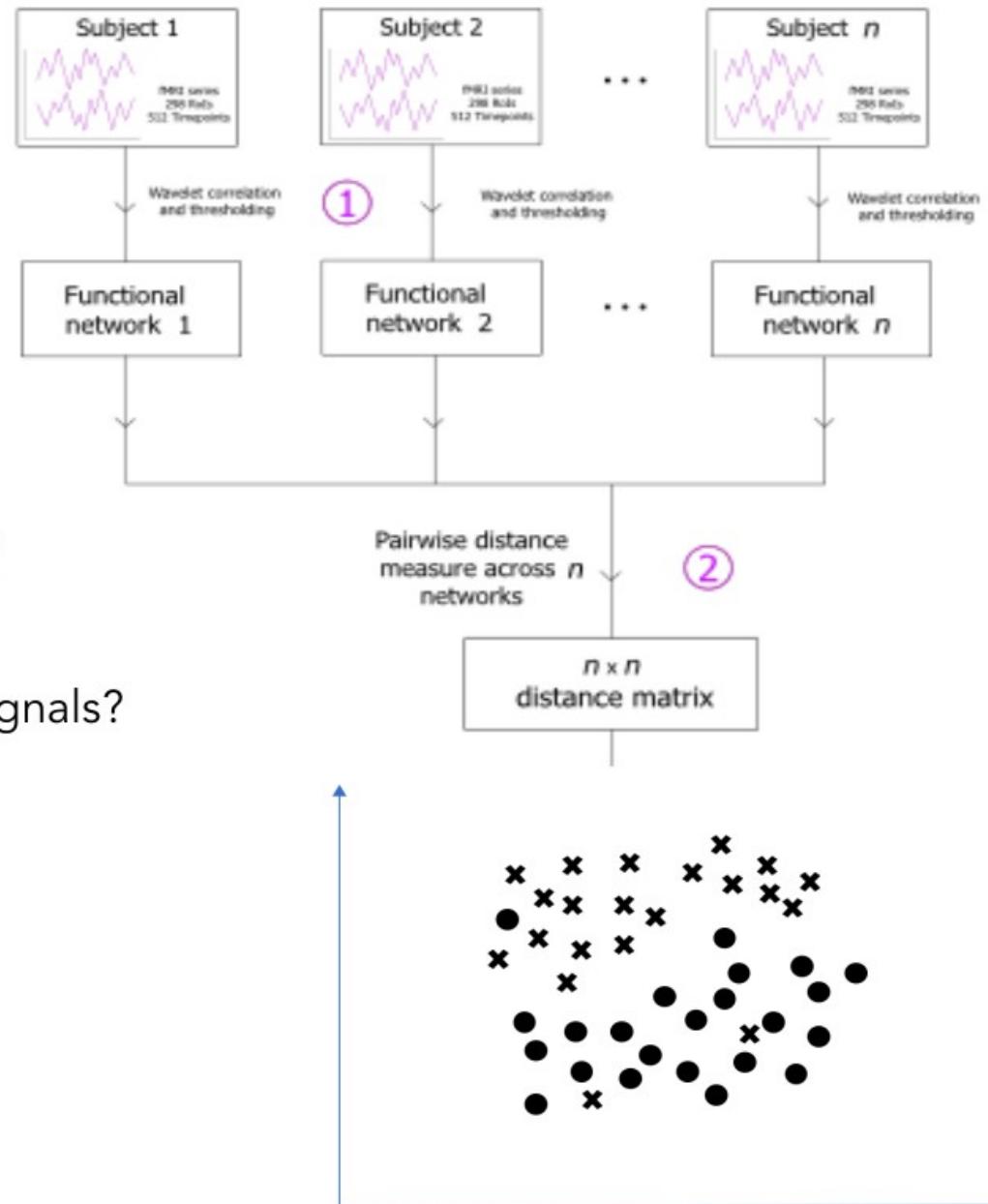


Bullmore, Sporns, *Nature Reviews Neuroscience* 10 (2009)

Correlation and Functional Networks

Typical study:
unsupervised clustering of diseases

Can we predict which subject have
schizophrenia by looking at brain signals?



Visibility Graphs

Visibility graphs were defined in computational geometry/computer science as the backbone graph capturing visibility paths (intervisible locations) in landscapes

- Each node represents a location
- Two locations are connected by a link if they are visible



Visibility Graphs

Visibility graphs were defined in computational geometry/computer science as the backbone graph capturing visibility paths (intervisible locations) in landscapes

- Each node represents a location
- Two locations are connected by a link if they are visible



Visibility Graphs

1D LANDSCAPES CAN BE CONSIDERED AS TIME SERIES



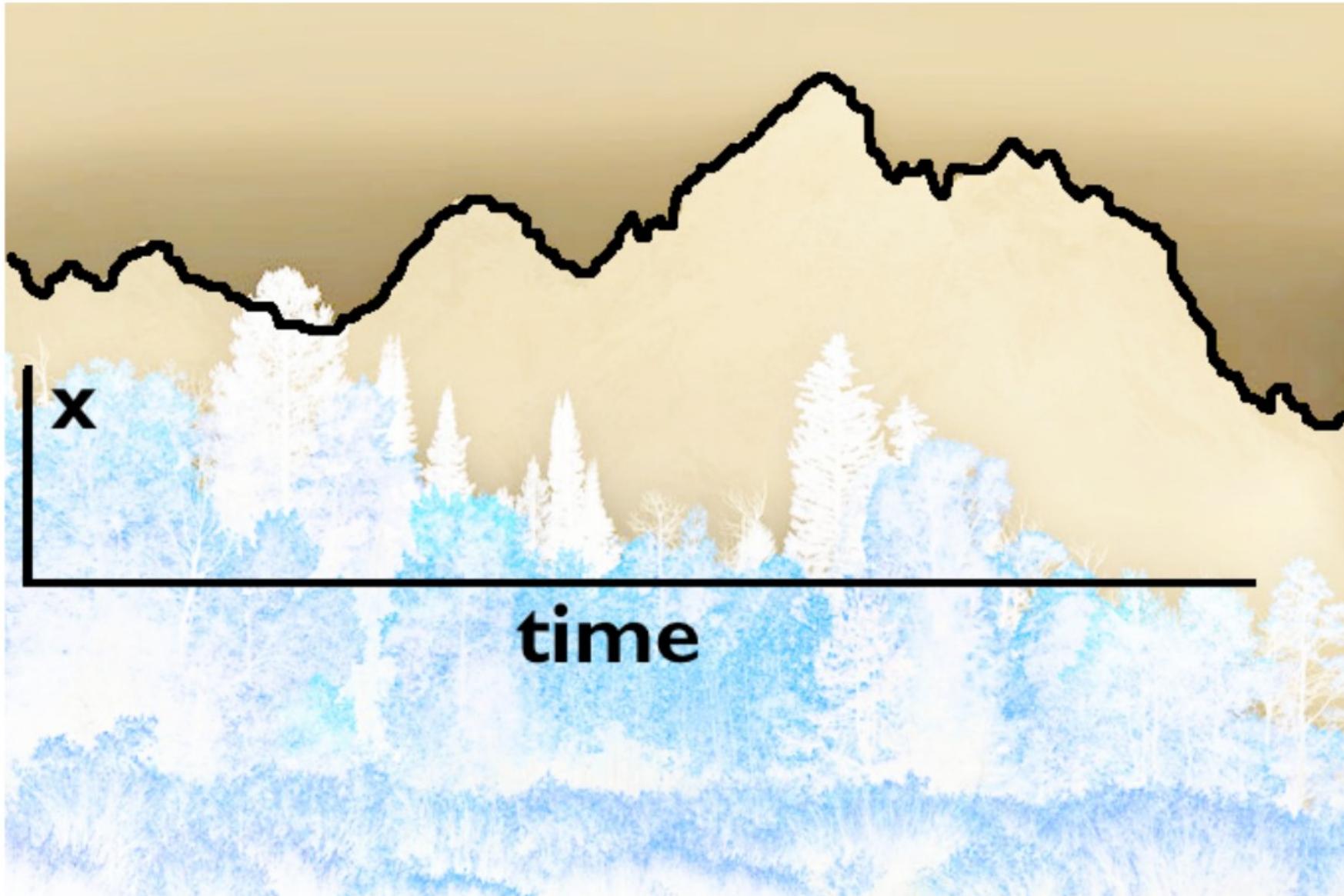
Visibility Graphs

1D LANDSCAPES CAN BE CONSIDERED AS TIME SERIES



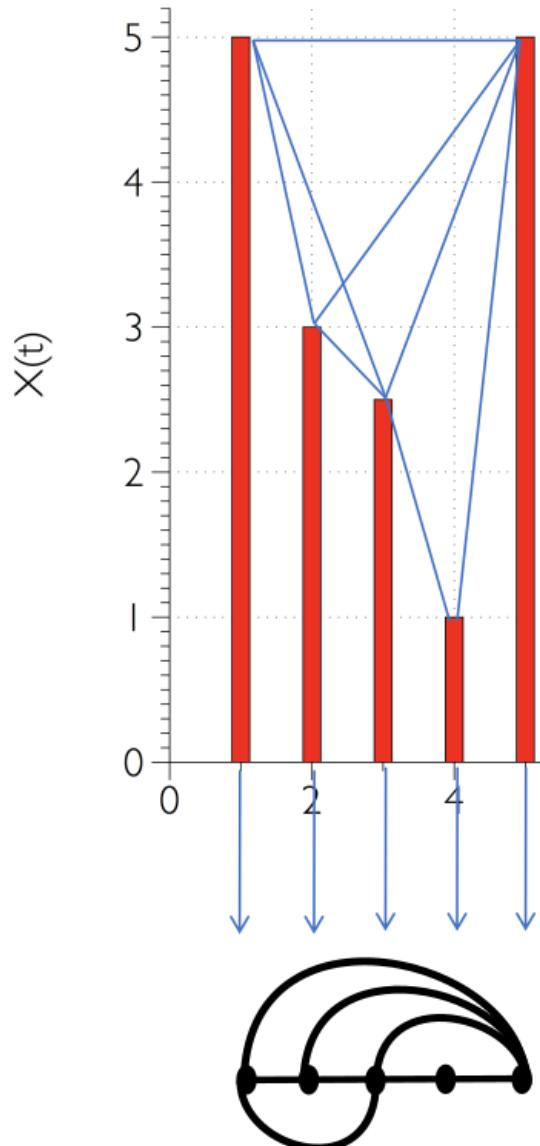
Visibility Graphs

1D LANDSCAPES CAN BE CONSIDERED AS TIME SERIES



Visibility Graphs

Natural Visibility Algorithm



For a time series of N data:

- * each datum is mapped into a node
- * two nodes are linked if a visibility criterion holds in the series

The resulting visibility graph:

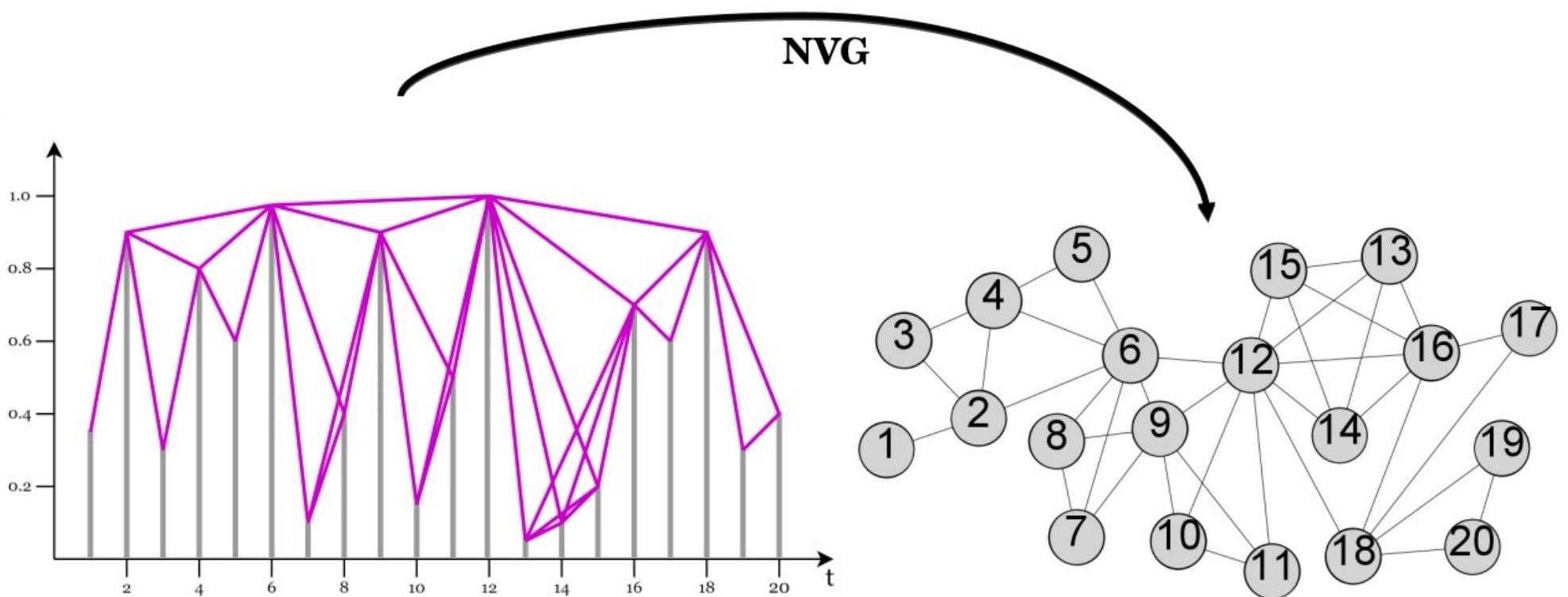
- * has N ordered nodes
- * is connected by a Hamiltonian path
- * is invariant under certain transformations in the series

Lacasa, Luque, Ballesteros, Luque, Nuño, PNAS 105 (2008)

Visibility Graphs

$$y_c = y_b + (y_a - y_b) \frac{(t_b - t_c)}{t_b - t_a}, \quad t_a < t_c < t_b$$

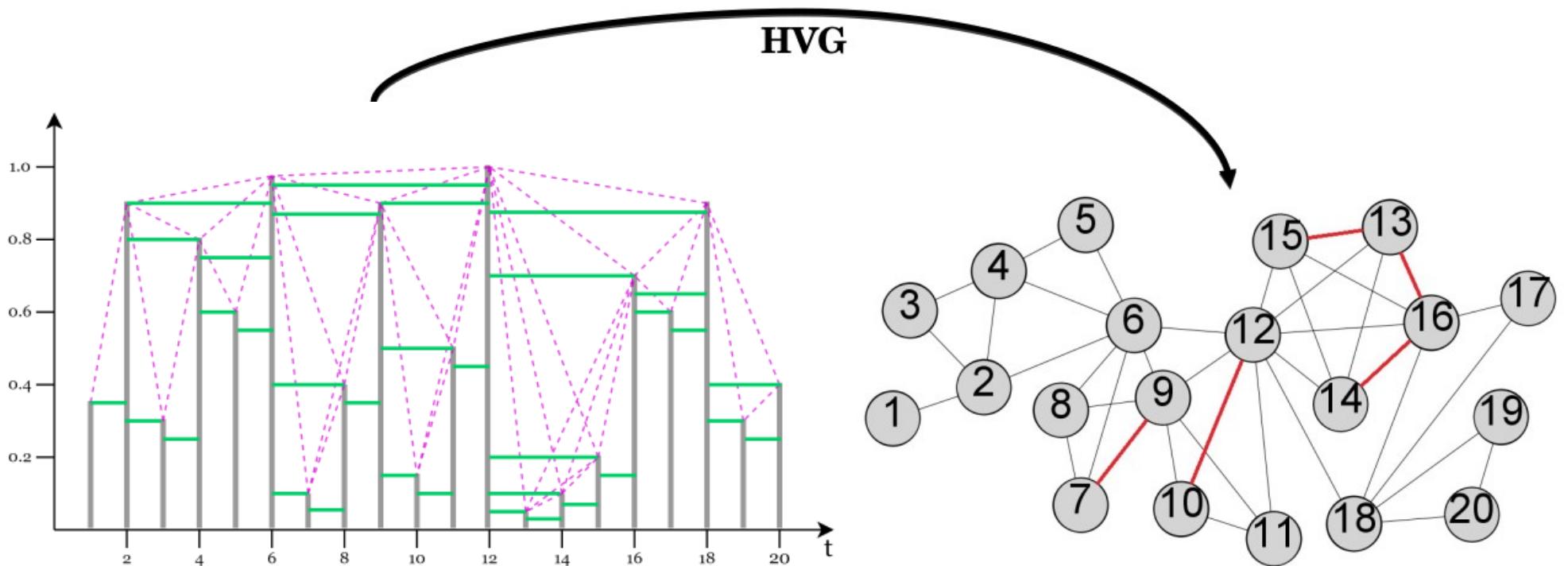
Natural Visibility Graph



Visibility Graphs

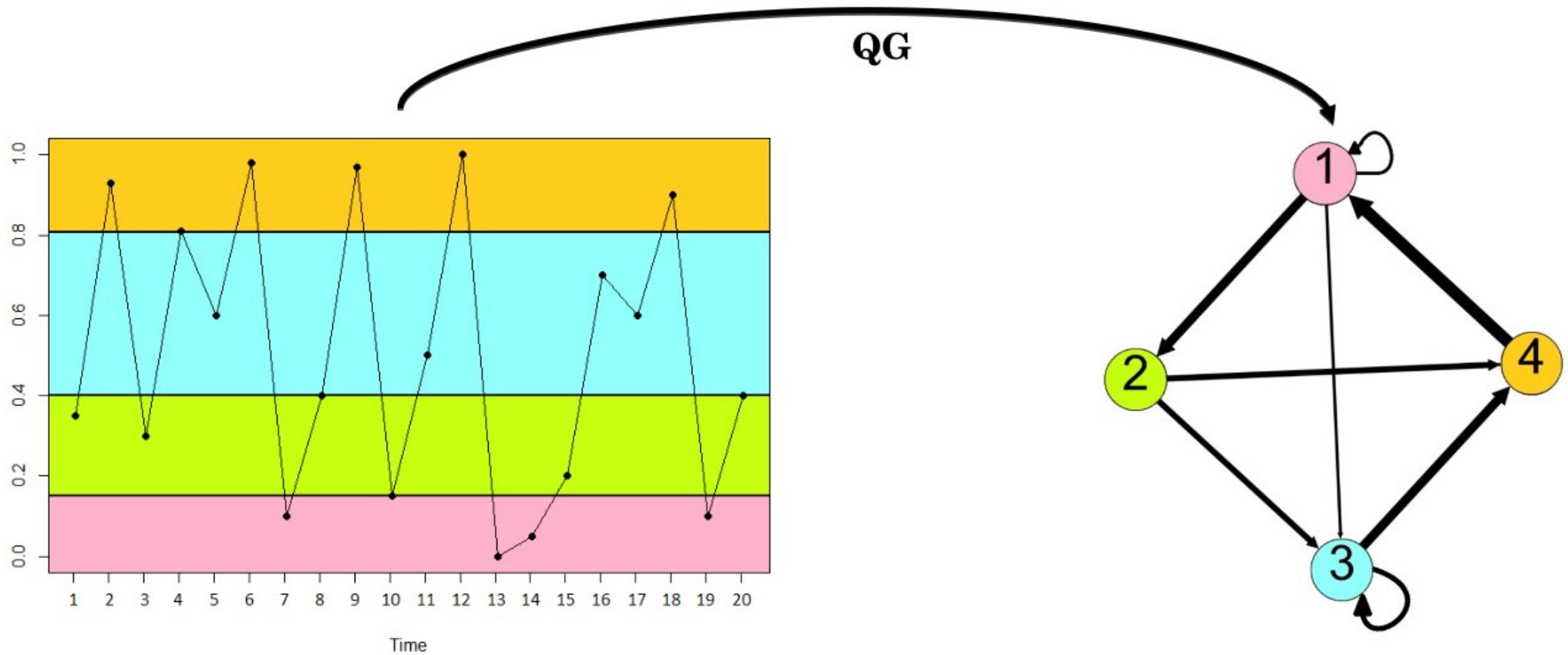
$$y_a, y_b > y_c, \quad t_a < t_c < t_b$$

Horizontal Visibility Graph



Quantile Graphs

Quantile Graph



Application: Time Series Clustering

Can simple topological measures of different networks distinguish different processes of time series?

*Vanessa Silva
MSc Thesis*

Time Series Clustering

- Distance-based methods
 - Similarity between observations
 - e.g. Dynamic Time Warping
- Characteristics-based methods
 - Similarity between global characteristics
 - e.g. trend, frequency, autocorrelation, Hurst
- Network-based methods
 - Similarity between topological measures
 - e.g. average degree, number of communities, clustering coefficient

Application: Time Series Clustering

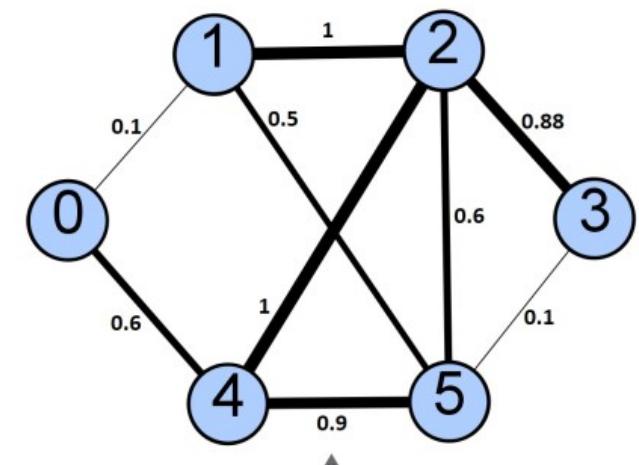
Can simple topological measures of different networks distinguish different processes of time series?

*Vanessa Silva
MSc Thesis*

Topological Metrics

- There is a vast set of topological metrics of graphs to study the particular characteristics of the system.

- **Average Degree (\bar{k})**
- **Average Path Length (\bar{d})**
- **Global Clustering Coefficient (C)**
- **Number of Communities (S)**
- **Modularity (Q)**



$$\begin{aligned}\bar{k} &= 1,89 \\ \bar{d} &= 1,47 \\ C &= 0,45 \\ S &= 2,00 \\ Q &= 0,05\end{aligned}$$

Application: Time Series Clustering

Can simple topological measures of different networks distinguish different processes of time series?

*Vanessa Silva
MSc Thesis*

Method

1. **Generate Complex Networks**
 - a. NVG, HVG, and QGs
2. **Calculate Metrics and Normalize**
 - a. \bar{k} , \bar{d} , C , S and Q
 - b. Min-Max normalization
3. **Dimensionality Reduction**
 - a. PCA and t-SNE
4. **Clustering Analysis**
 1. k-means

Application: Time Series Clustering

Can simple topological measures of different networks distinguish different processes of time series?

Vanessa Silva
MSc Thesis

Time Series Models

- White Noise (i.i.d)

- Linear models

- AR(1)

- Smoother

- AR(2)

- Pseudo-Periodic

- ARIMA

- Stochastic Trend

- ARFIMA

- Long Memory

- Nonlinear models

- SETAR

- Regimes

- HMM

- States

- INAR

- Integer Valued Data

- GARCH

- Conditional
Heterocesdaticity
and Asymmetry

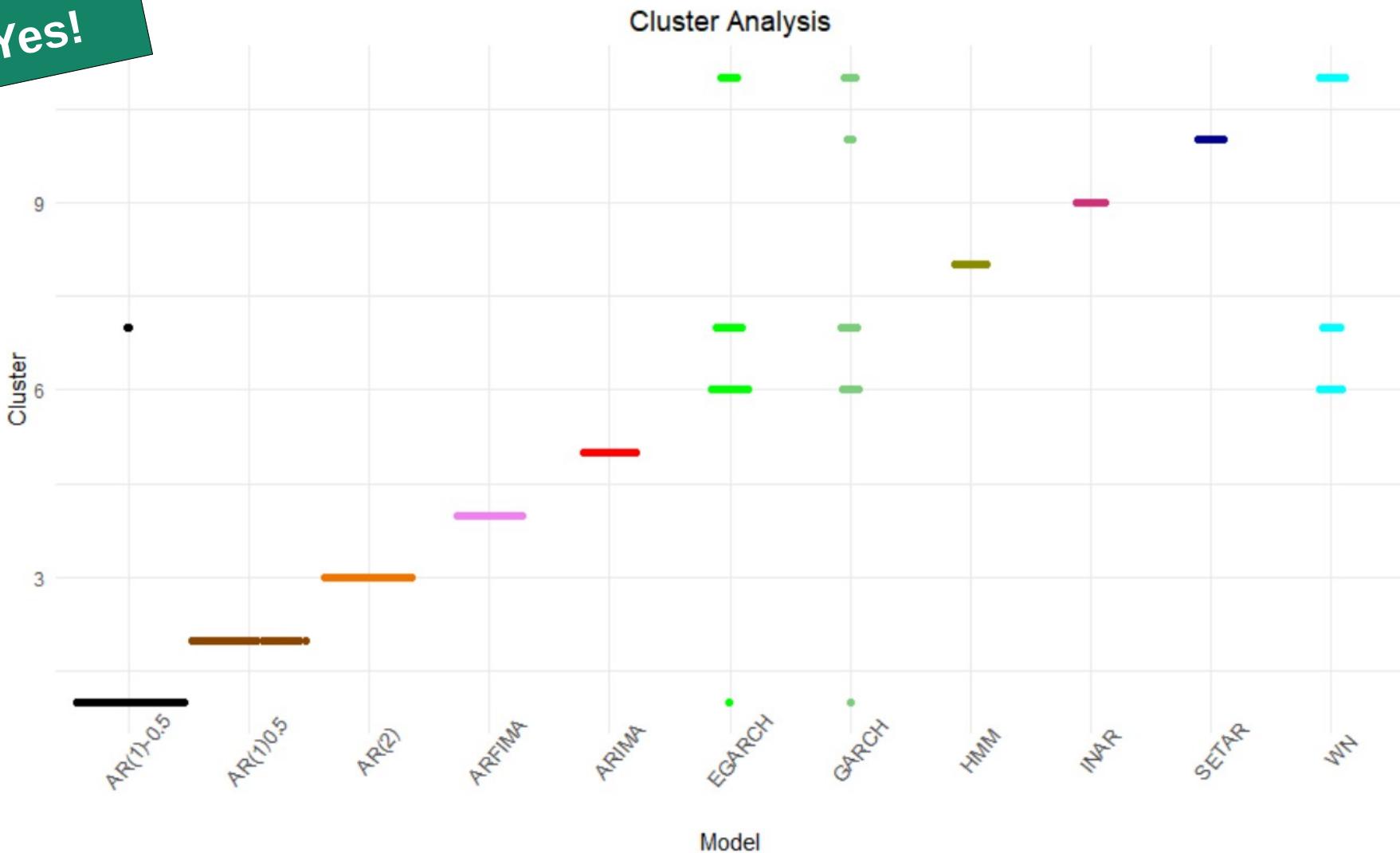
- EGARCH

Create randomized instances of each of these models

Application: Time Series Clustering

Can simple topological measures of different networks distinguish different processes of time series?

Yes!

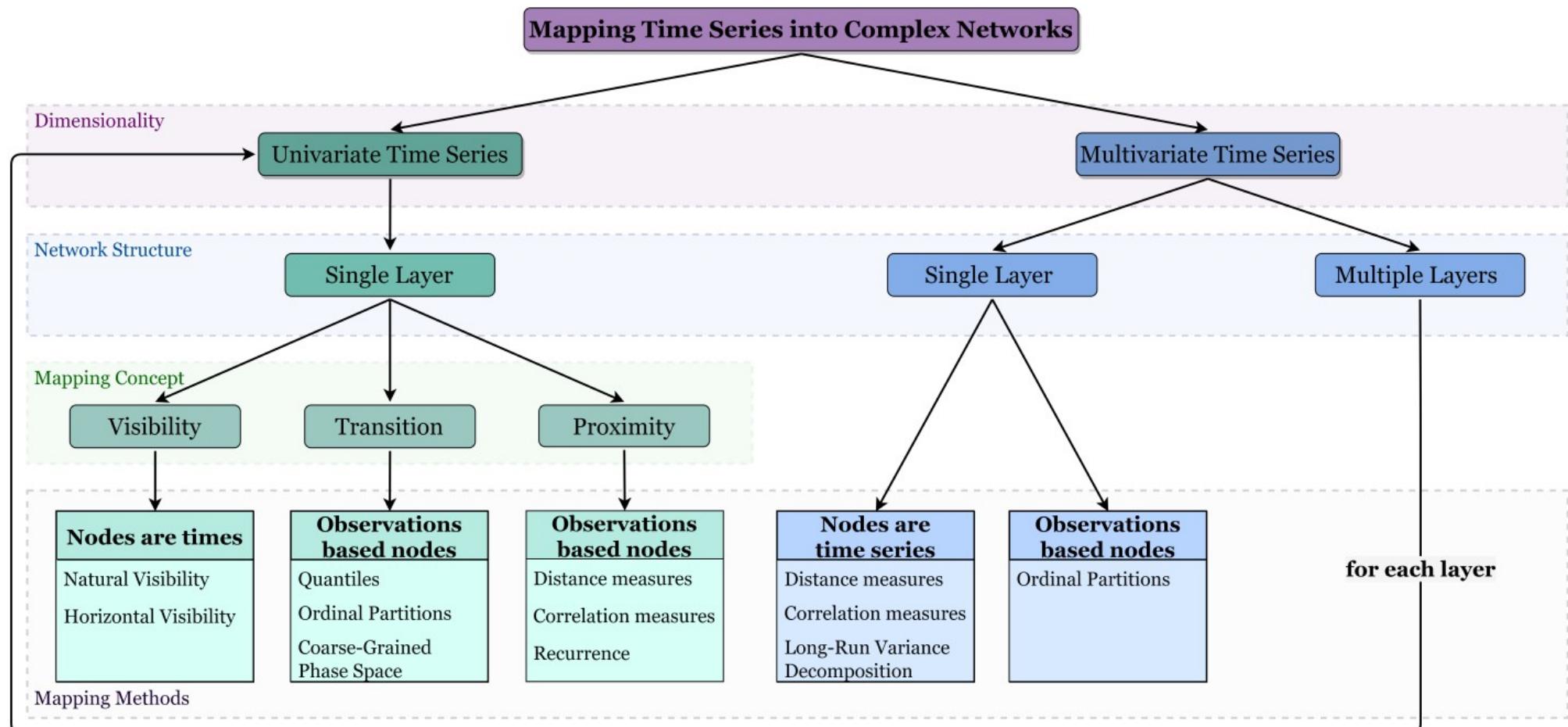


Vanessa Silva
MSc Thesis

More on TSA via NetSci

Time Series Analysis via Network Science: Concepts and Algorithms*

Vanessa Freitas Silva¹, Maria Eduarda Silva², Pedro Ribeiro¹, and Fernando Silva¹



Extensions

International Journal of Data Science and Analytics
<https://doi.org/10.1007/s41060-024-00561-6>

REGULAR PAPER

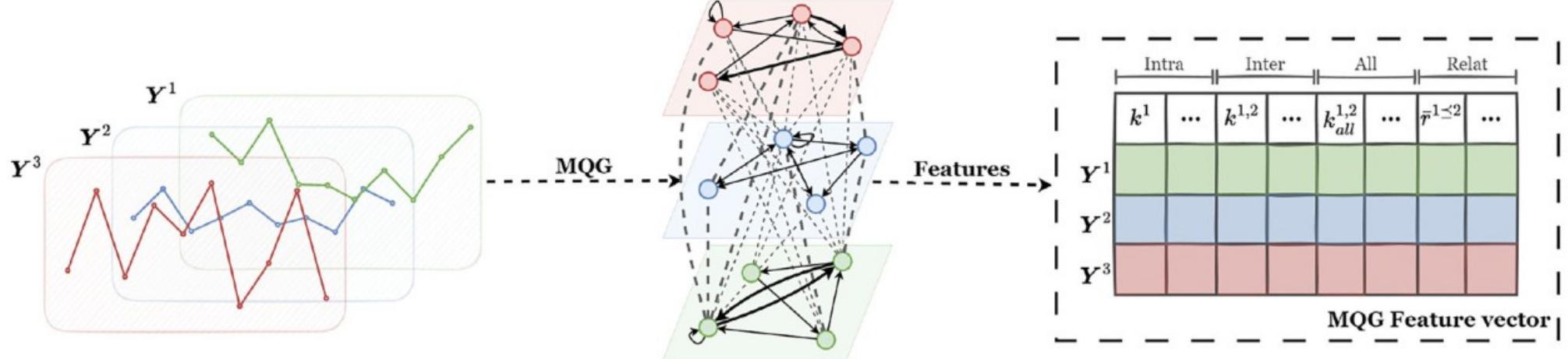
Vanessa Silva
PhD Thesis



Multilayer quantile graph for multivariate time series analysis and dimensionality reduction

Vanessa Freitas Silva¹ · Maria Eduarda Silva² · Pedro Ribeiro¹ · Fernando Silva¹

Received: 2 October 2023 / Accepted: 6 May 2024
© The Author(s) 2024

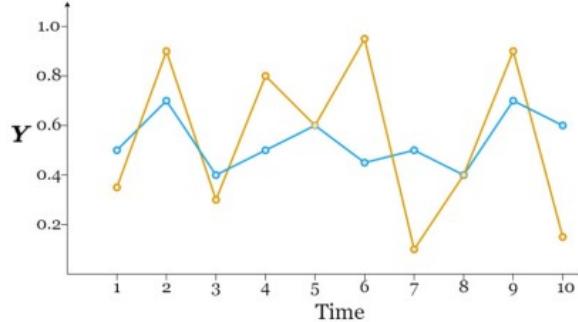


Extensions

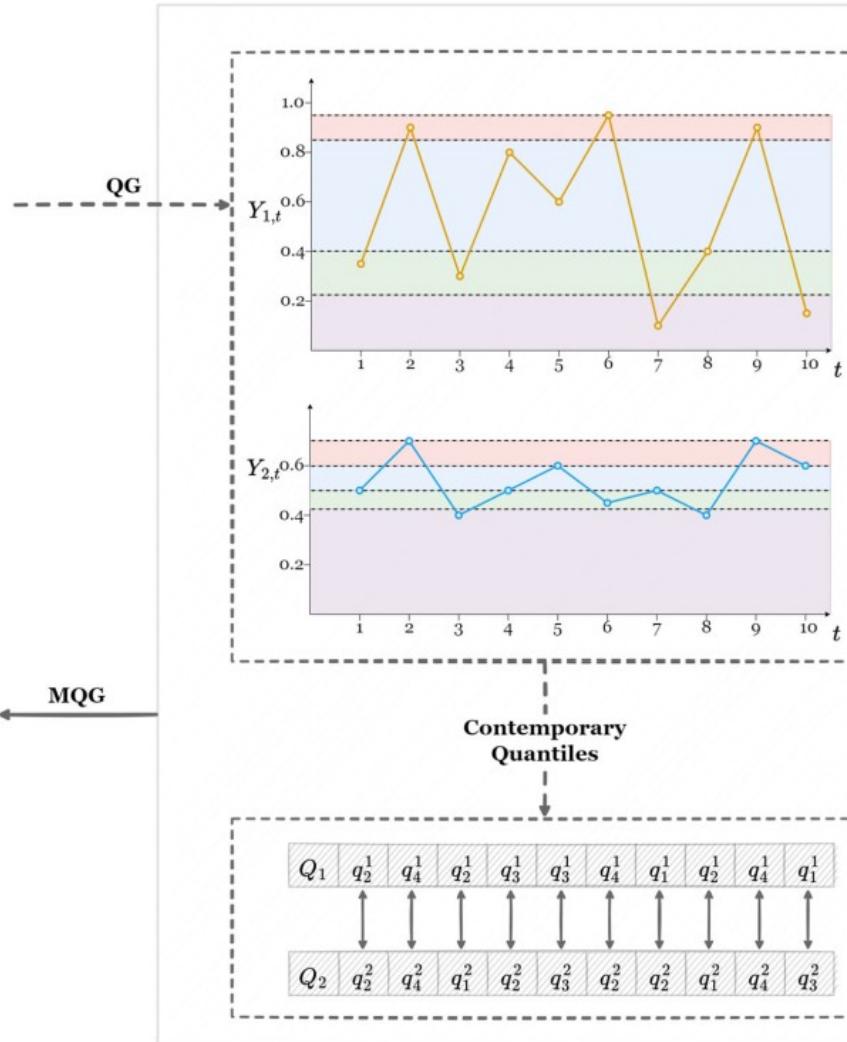
*Vanessa Silva
PhD Thesis*

Multilayer quantile graph for multivariate time series analysis and dimensionality reduction

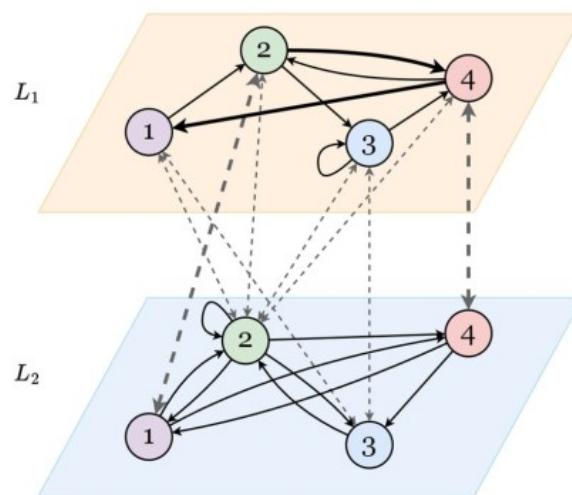
(a)



(b)



(c)



Extensions

*Vanessa Silva
PhD Thesis*

Data Mining and Knowledge Discovery (2025) 39:17
<https://doi.org/10.1007/s10618-025-01089-4>



Multilayer horizontal visibility graphs for multivariate time series analysis

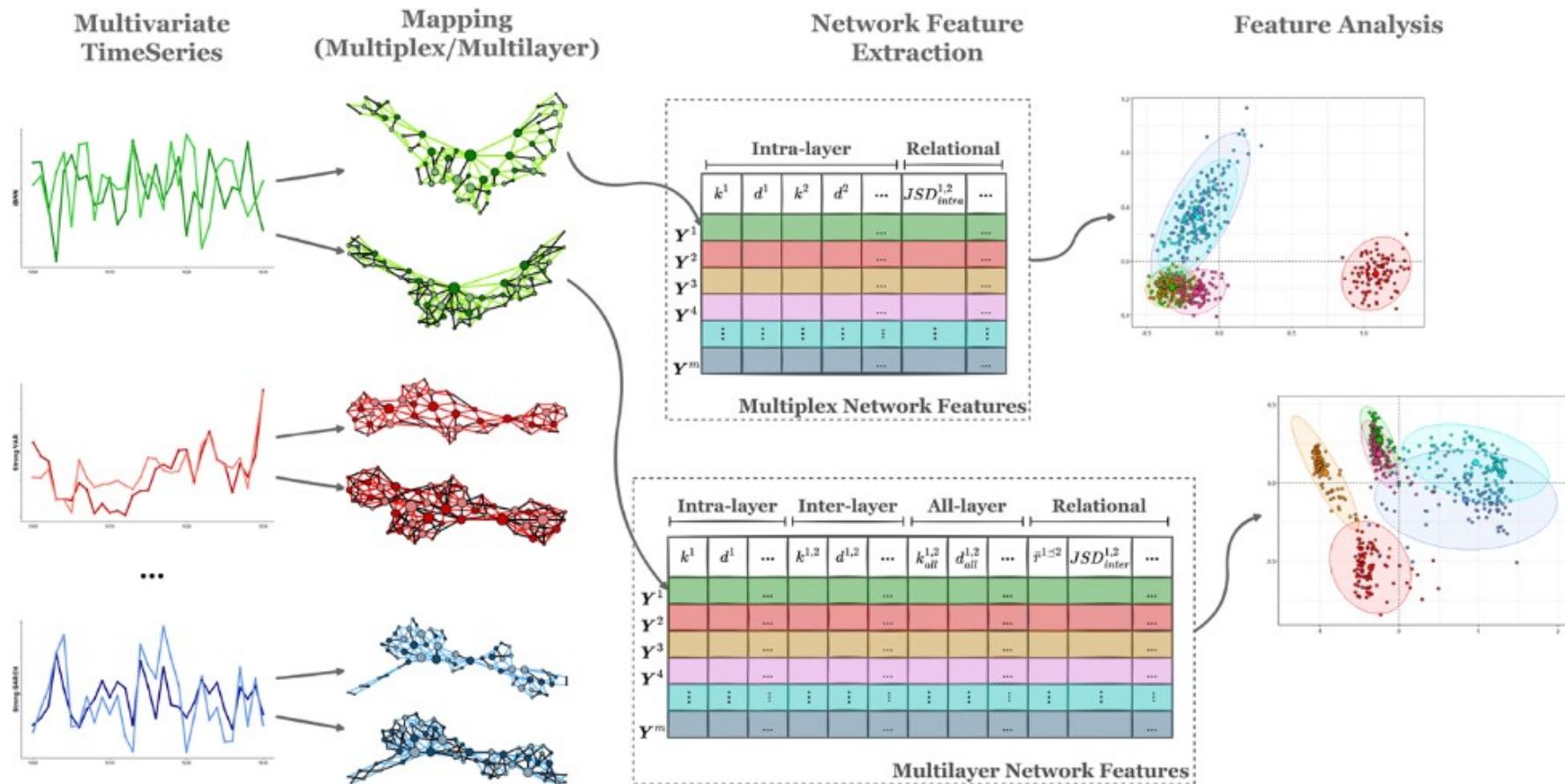
Vanessa Freitas Silva¹ · Maria Eduarda Silva² · Pedro Ribeiro¹ · Fernando Silva¹

Received: 6 February 2024 / Accepted: 4 January 2025 / Published online: 3 March 2025
© The Author(s) 2025

Extensions

Vanessa Silva
PhD Thesis

Multilayer horizontal visibility graphs for multivariate time series analysis

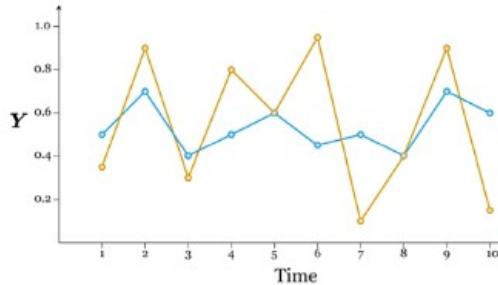


Extensions

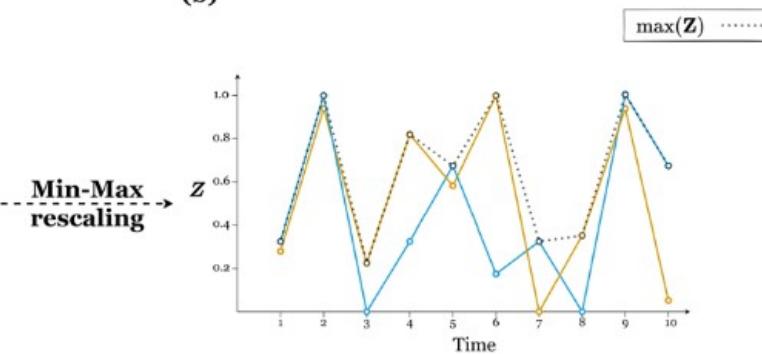
Vanessa Silva
PhD Thesis

Multilayer horizontal visibility graphs for multivariate time series analysis

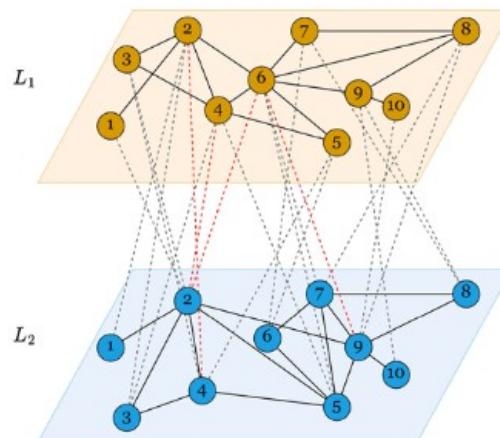
(a)



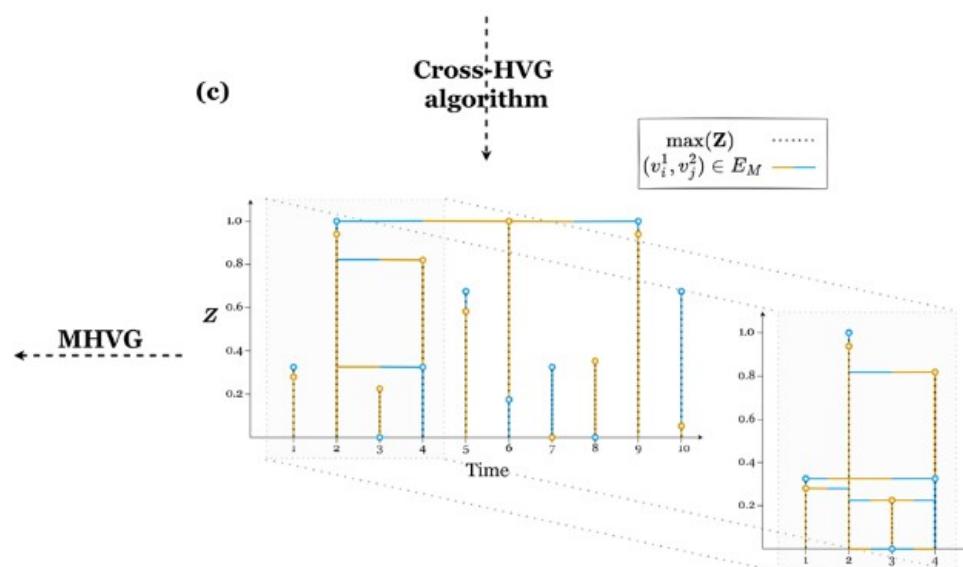
(b)



(d)



(c)



Vanessa Silva's Work

Time Series Analysis based on Complex Networks

Vanessa Alexandra Freitas da Silva

Master's degree in Networks and Informatics Systems Engineering
Computer Science Department
2018

Supervisor

Fernando Manuel Augusto da Silva, Full Professor, Faculty of Sciences, University of Porto

Co-supervisor

Pedro Manuel Pinto Ribeiro, Assistant Professor, Faculty of Sciences, University of Porto

Co-supervisor

Maria Eduarda da Rocha Pinto Augusto da Silva, Associate Professor, Faculty of Economics, University of Porto



Multidimensional Time Series Analysis: A Complex Networks Approach

Vanessa Alexandra Freitas da Silva

Doctoral Program in Computer Science of the Universities of Minho, Aveiro and Porto (MAPi)
Computer Science Department
2023



Time series forecasting via Network Science

Filipe Godinho Justiça

Master's degree in Data Science
Computer Science Department
2022

Supervisor

Pedro Manuel Pinto Ribeiro, Assistant Professor, Faculty of Science, University of Porto

Co-supervisor

Maria Eduarda da Rocha Pinto Augusto da Silva, Associate Professor, Faculty of Economics, University of Porto

