Data mining in large graphs

Christos Faloutsos
Carnegie Mellon University
www.cs.cmu.edu/~christos

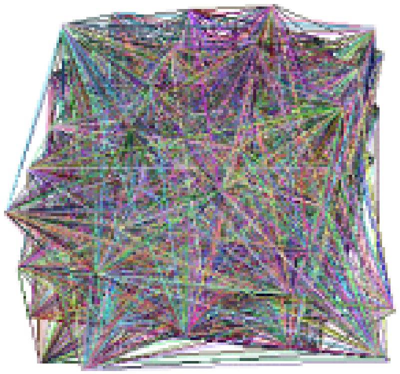
Outline

- Introduction - motivation
- Patterns & Power laws
- Scalability & Fast algorithms
- Fractals, graphs and power laws
- Conclusions

Introduction

- How do real networks look like?
- Any ‘laws’/patterns they obey?
- How to handle huge graphs?

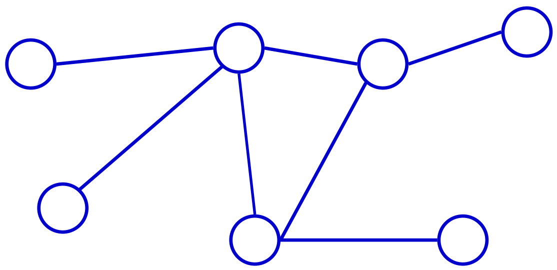
Problem #1 - network and graph mining



- How does the Internet look like?
- How does the web look like?
- What constitutes a 'normal' social network?
- What is the 'market value' of a customer?
- In a food web, which gene/species affects the others the most?

Problem#1: Patterns

Given a graph:



- which node to market-to / defend / immunize first?
- Are there un-natural sub-graphs? (criminals' rings or terrorist cells)?
- How do peer-to-peer (P2P) networks evolve?

Problem #2: Scalability

- How to handle huge graphs (>>10**5 nodes)

Solutions

- Problem#1 - patterns: New tools: power laws, self-similarity and ‘fractals’ work, where traditional assumptions fail
- Problem#2 - scalability: Approximations

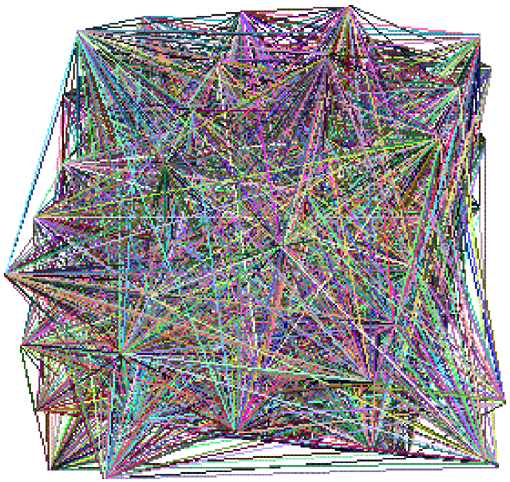
In detail:

Outline

- Introduction - motivation
- **Patterns & Power laws**
- Scalability & Fast algorithms
- Fractals, graphs and power laws
- Conclusions

Problem #1 - topology

How does the Internet look like? Any rules?

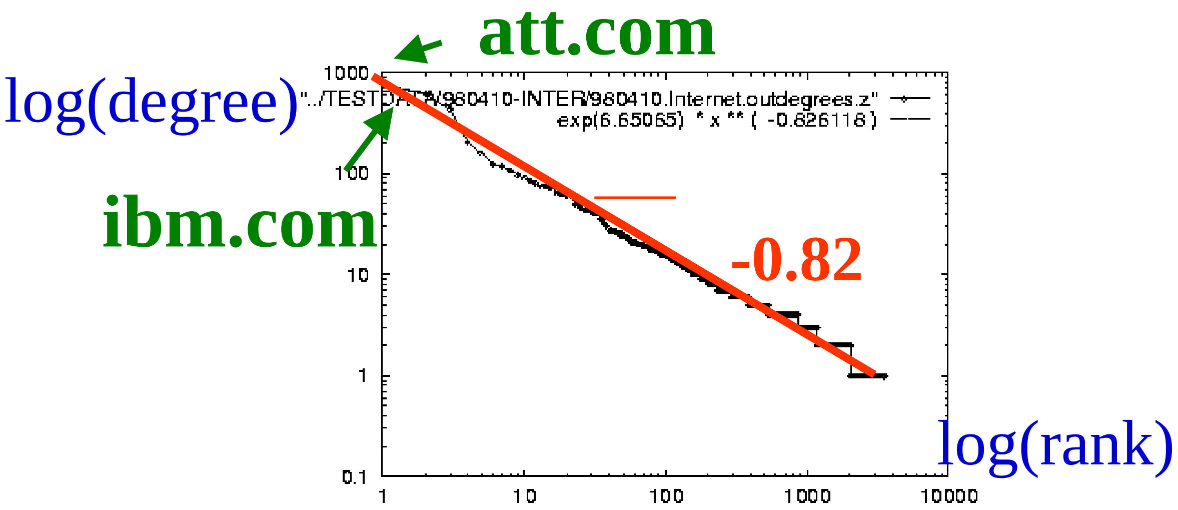


A: self-similarity and power-laws!

Solution#1:

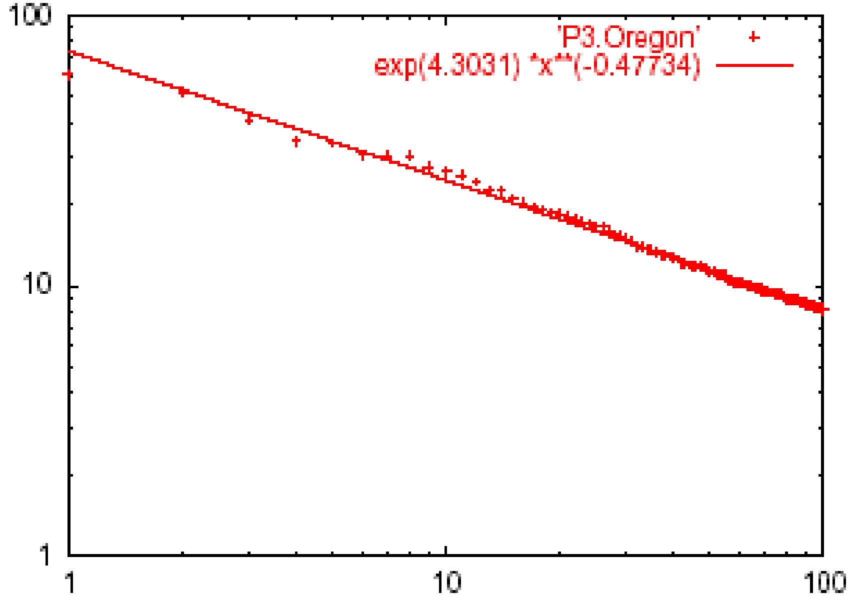
- A1: Power law in the degree distribution [SIGCOMM99]

internet domains



Solution#1’: Eigen Exponent *E*

Eigenvalue



Exponent = slope

E = *-0.48*

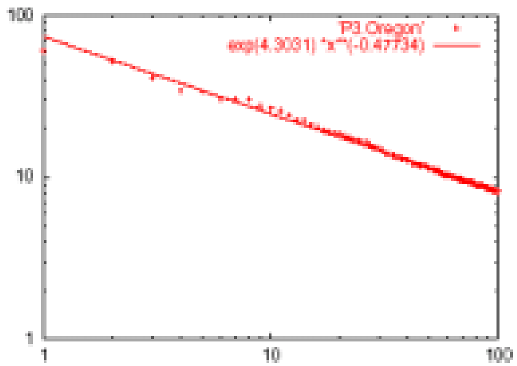
May 2001

Rank of decreasing eigenvalue

- A2: power law in the eigenvalues of the adjacency matrix

Solution#1': Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Explanation [Mihail & Papadimitriou, 2002]:

$$E = R/2 \qquad \text{(!!)}$$

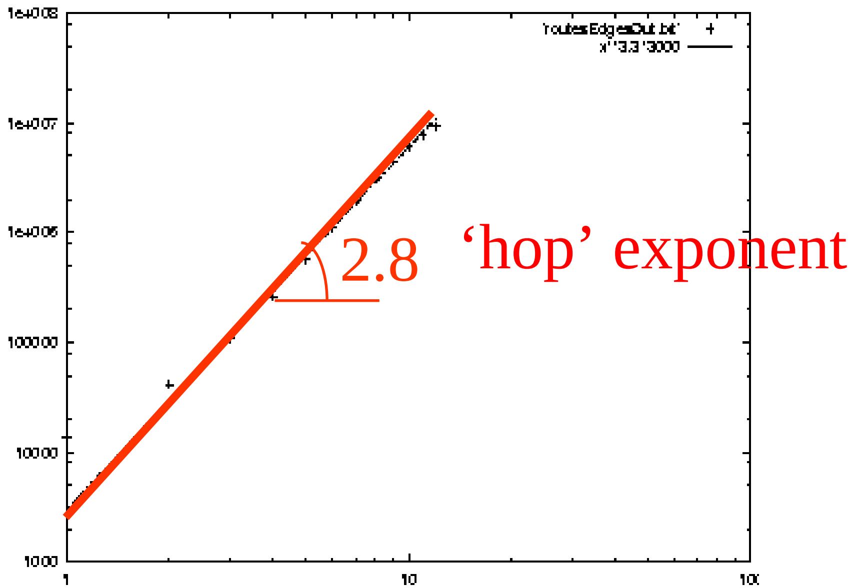
(because, in a forest of 'stars', $\lambda_i \sim sqrt(degree_i)$)

Solution#1'': Hop Exponent *H*

- A3: neighborhood function *N(h)* = number of pairs within *h* hops or less - power law, too!

log(#pairs)

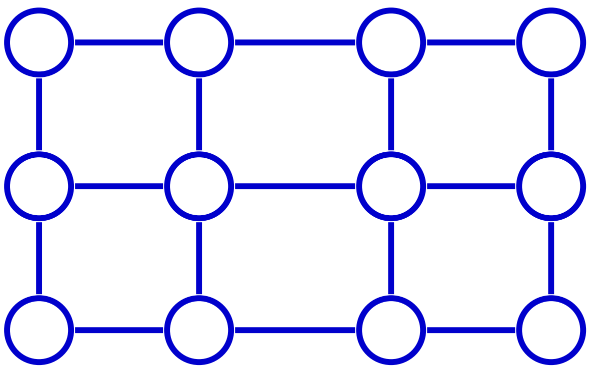
internet



log(hops)

Hop exp. = 1

Hop exp. = 2



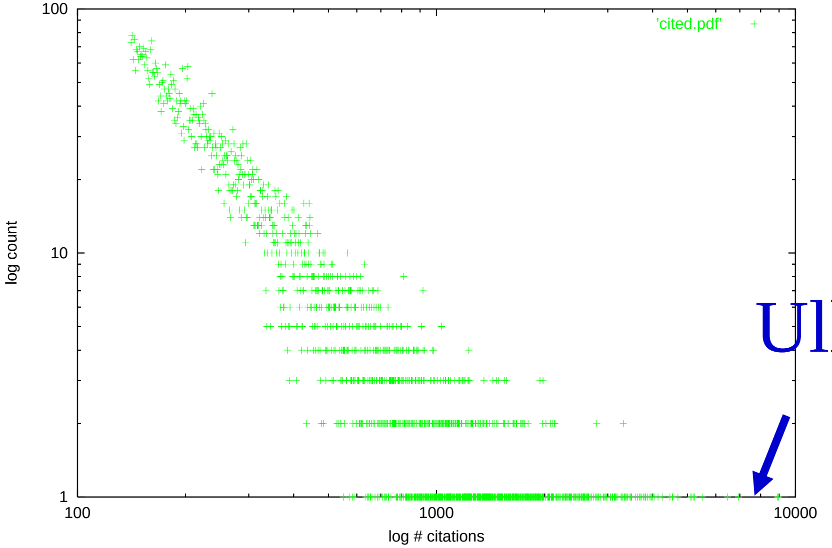
But:

- Q1: How about graphs from other domains?
- Q2: How about temporal evolution?

Q1: More power laws:

citation counts: (*citeseer.nj.nec.com* 6/2001)

log(count)

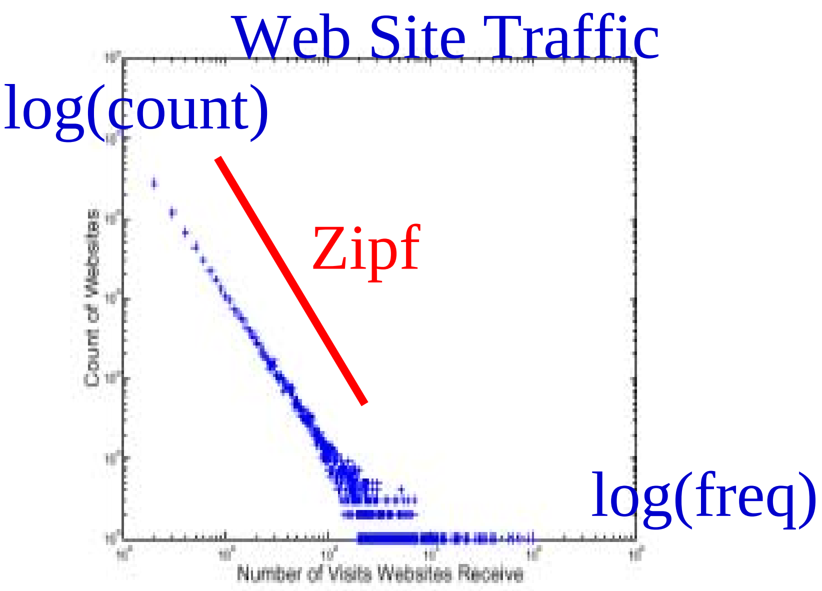


Ullman

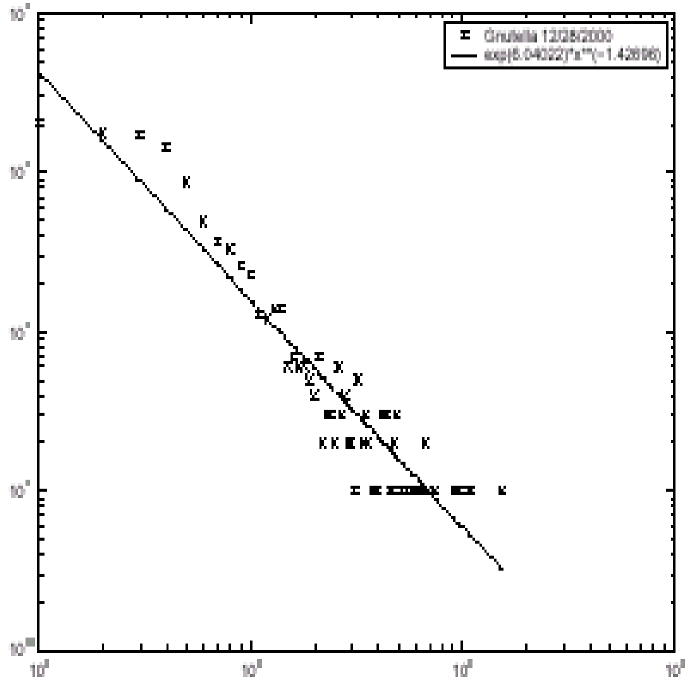
log(#citations)

Q1: More power laws:

- web hit counts [w/ A. Montgomery]



Q1: The Peer-to-Peer Topology



(a) Gnutella snapshot from Dec. 28, 2000 (|r|=0.94)

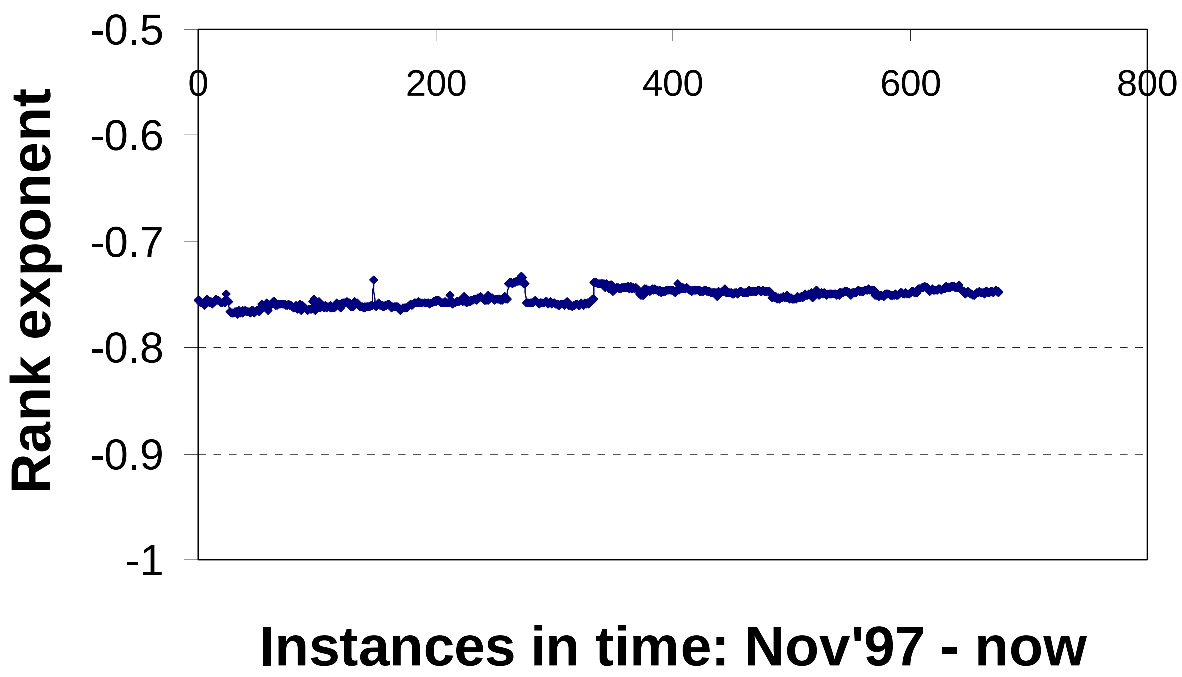
[Jovanovic+]

- Frequency versus degree
- Number of adjacent peers follows a power-law

Q1: More Power laws

- Also hold for other web graphs [Barabasi+], [Broder+], with additional ‘rules’ (bi-partite cores follow power laws)

Q2: Time Evolution: rank *R*



Domain level

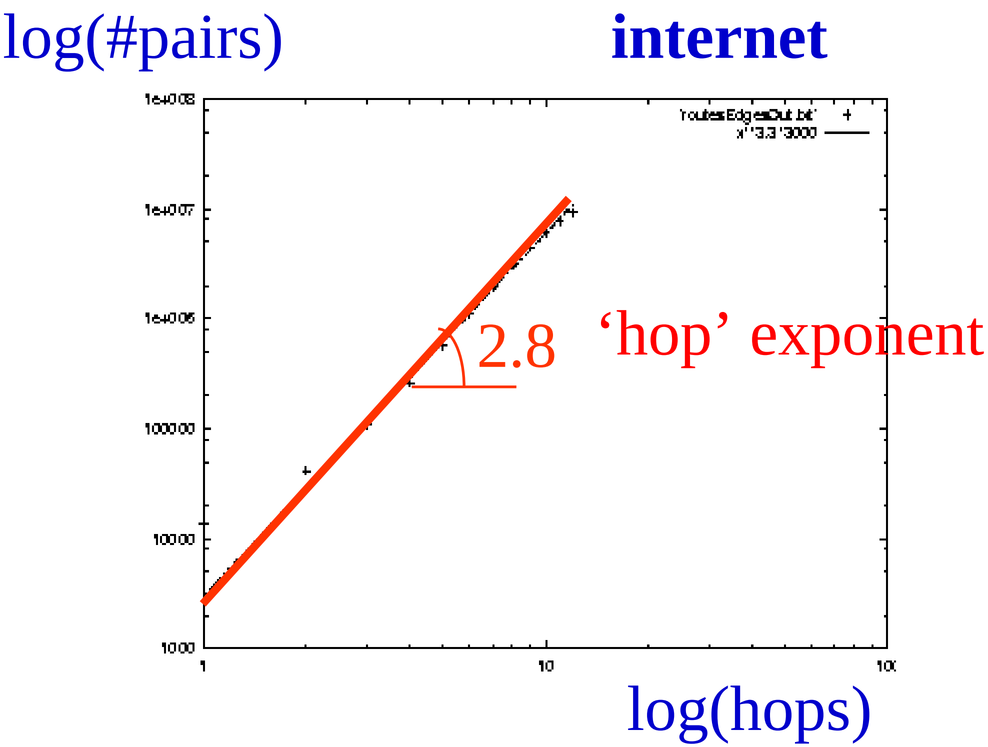
- The rank exponent has not changed!

Outline

- Introduction - motivation
- Patterns & Power laws
- **Scalability & Fast algorithms**
- Fractals, graphs and power laws
- Conclusions

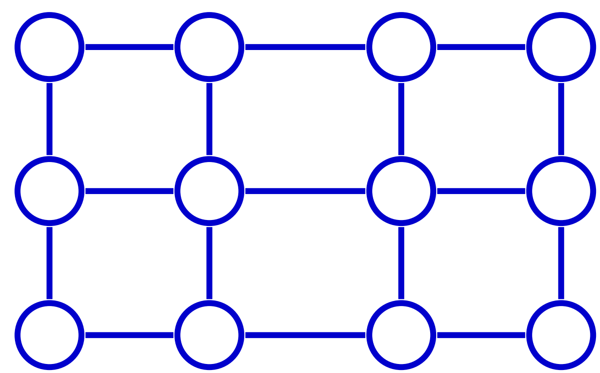
Hop Exponent *H*

- A3: neighborhood function *N(h)* = number of pairs within *h* hops or less - power law, too!



Hop exp. = 1

Hop exp. = 2



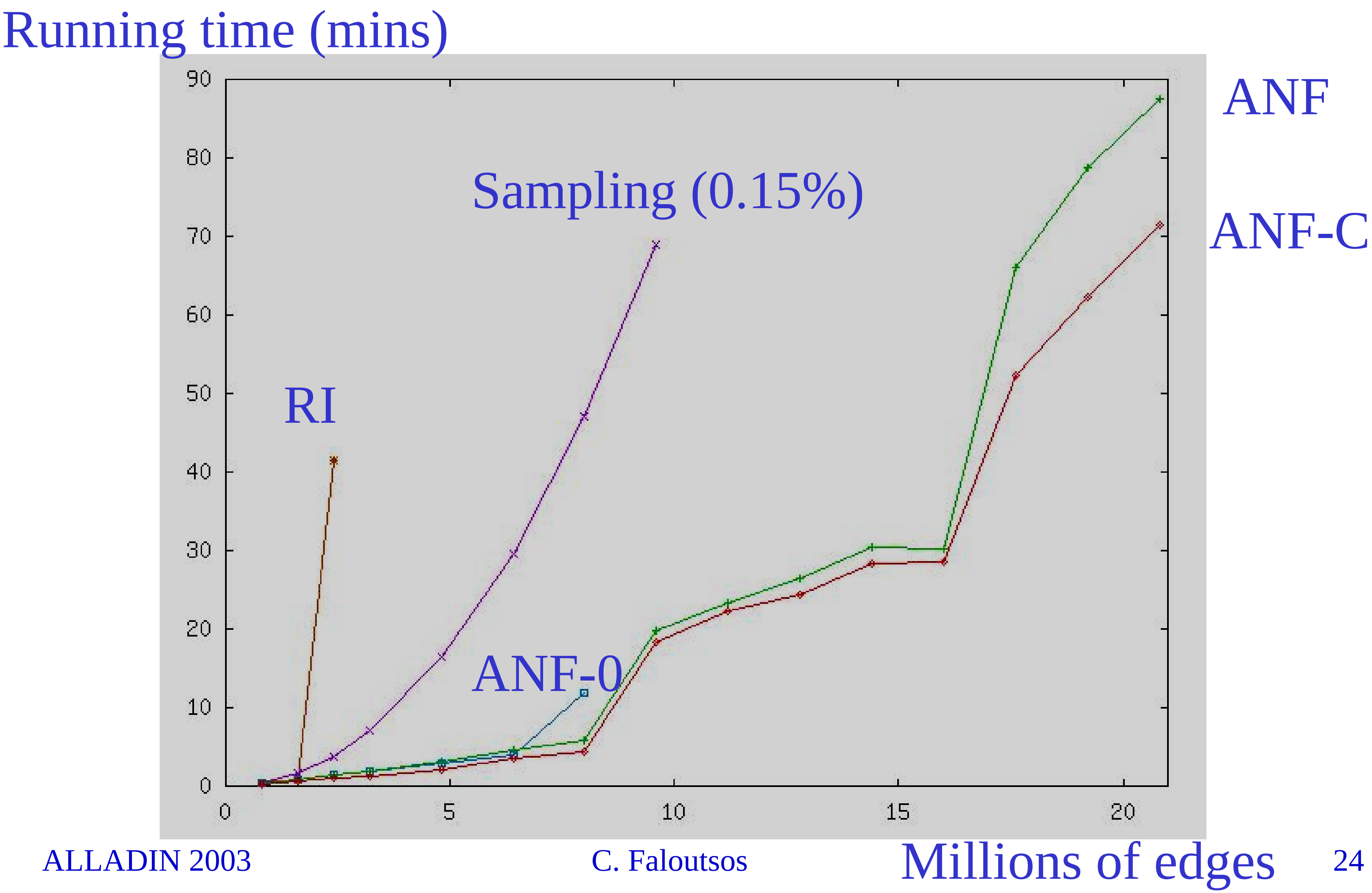
More on the hop exponent

- 'Intrinsic'/fractal dimensionality of the nodes of the graph
- But: naively it needs O(N**2) (terrible for large graphs)
- What to do?

Solution:

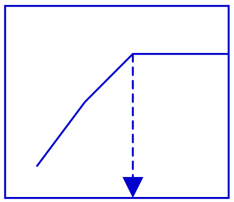
- Approximation: 'ANF' (approx. neighborhood function [KDD02, w/ C. Palmer and P. Gibbons] - response time: from **day** to **minutes**

Scalability of ANF!



(Approx.) neighborhood function

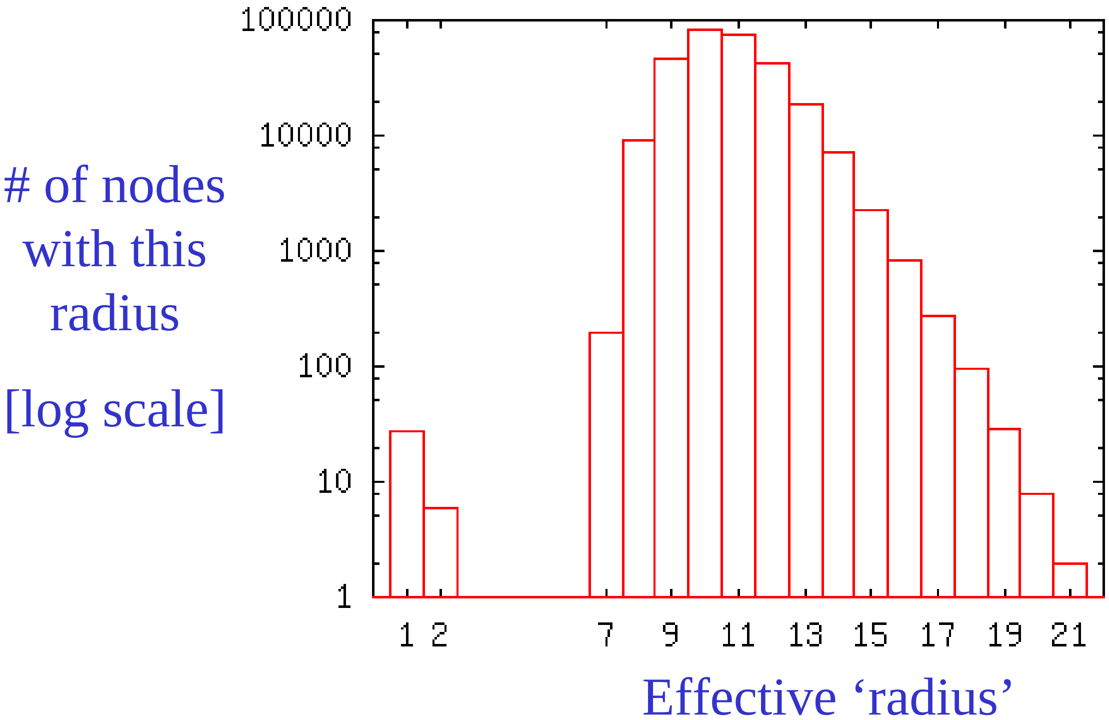
N(h)



- Useful for estimating the diameter of a graph;
- the ``effective radius'' of a node (distance to 90%-tile of the other nodes)
- the connectivity under failures
- quick checks for (dis-)similarity between two graphs

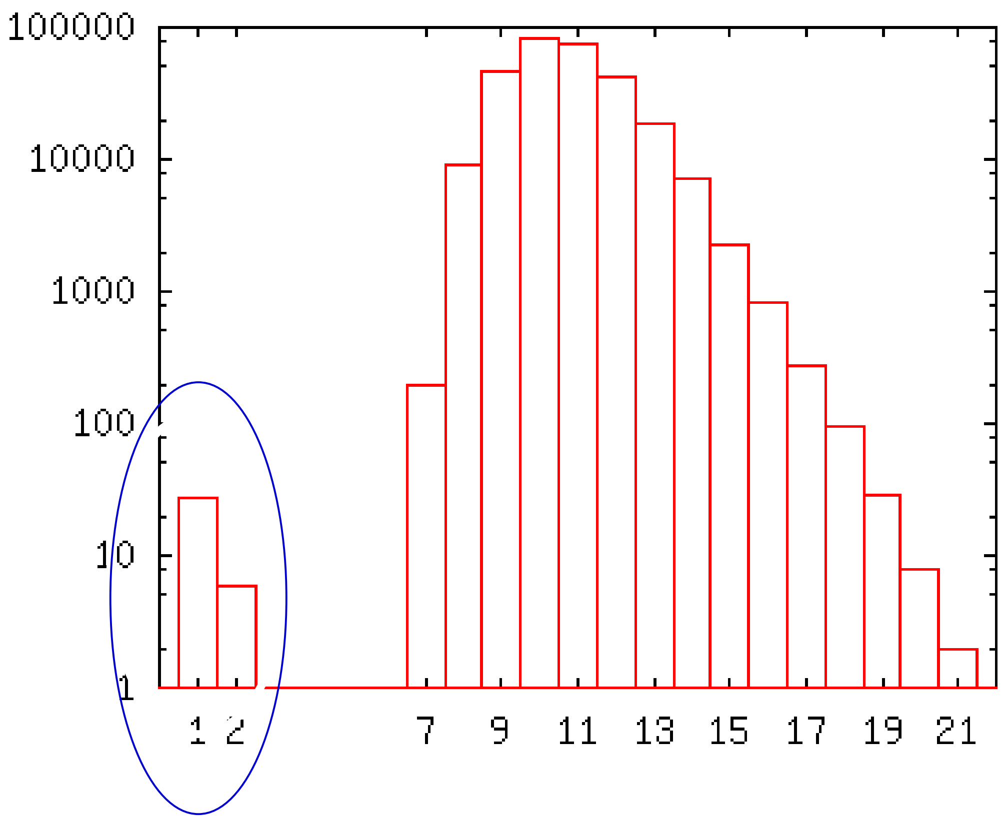
Effective Radius

- Effective Radius(x): radius that covers 90% of total nodes, starting from node 'x'

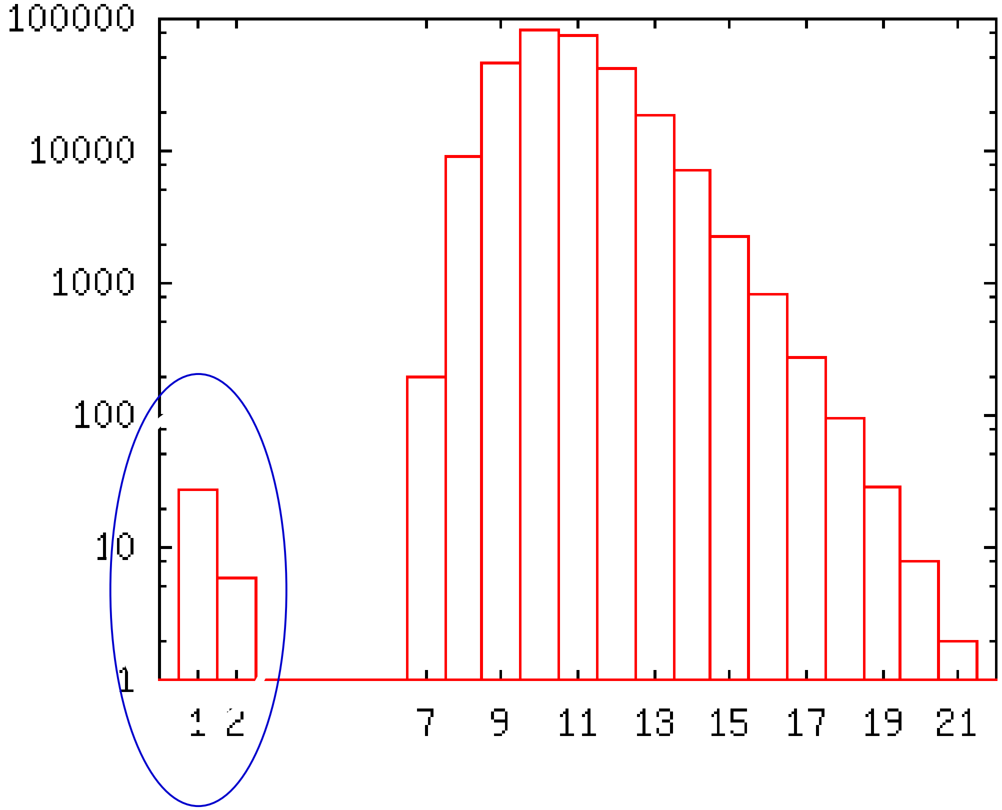


We can learn a lot by looking at the different parts of this histogram

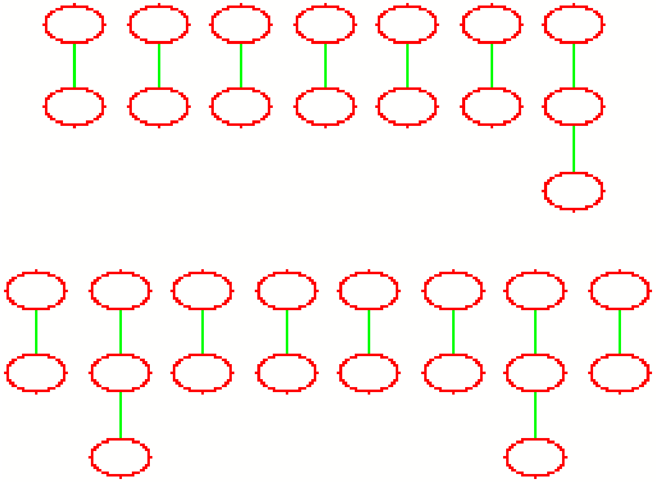
Small radii - explanation?



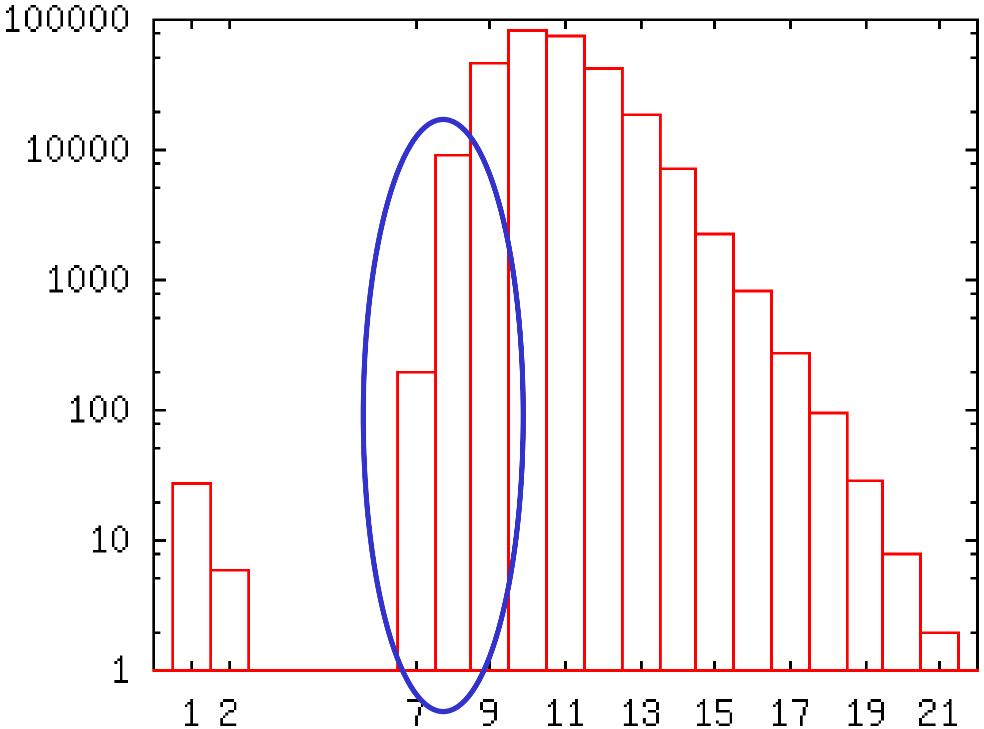
Identify Outliers / Data Errors



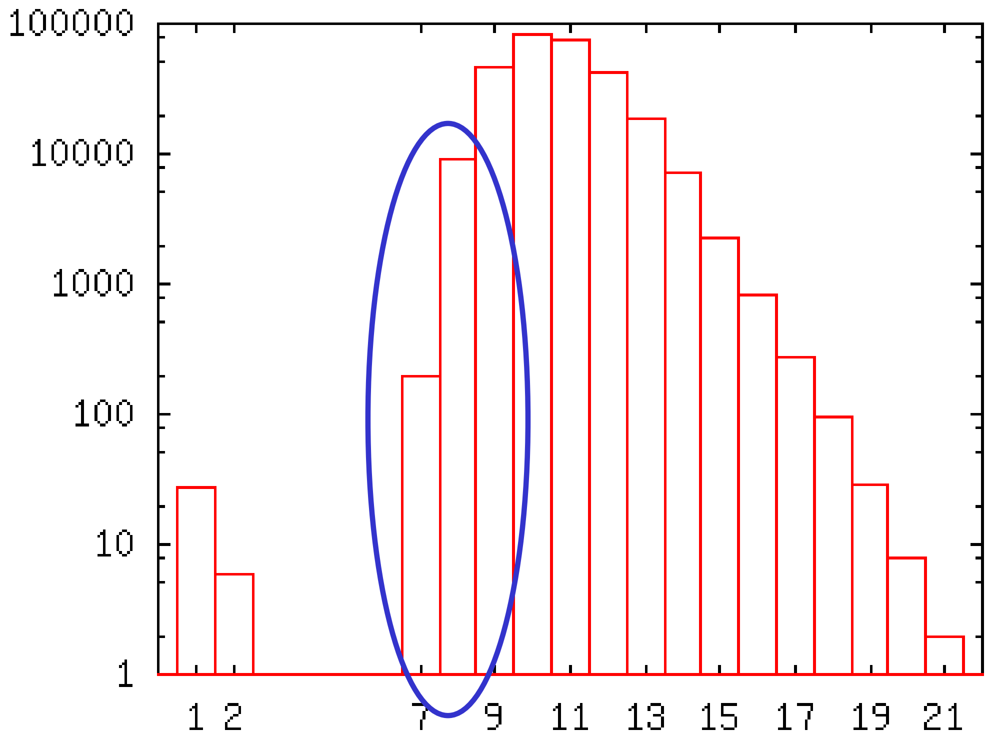
Actual Subgraph of these nodes



Nodes of radius 7-9?

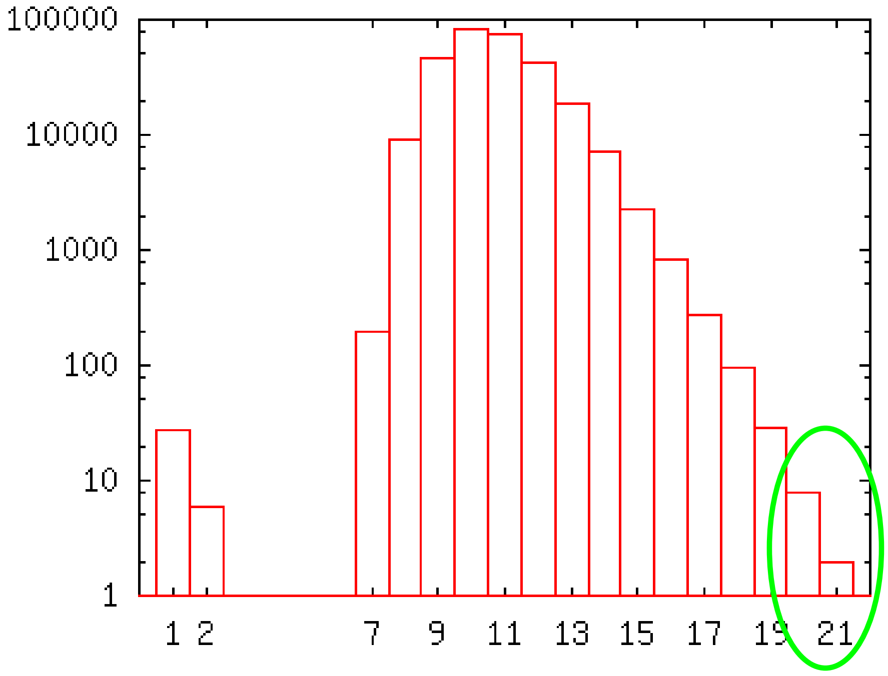


Identify "Important" Nodes

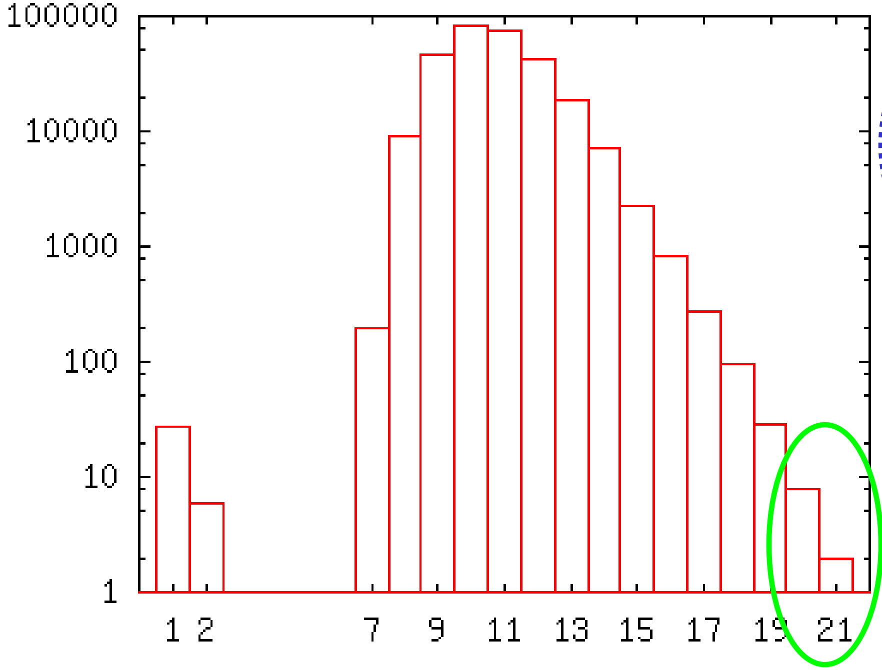


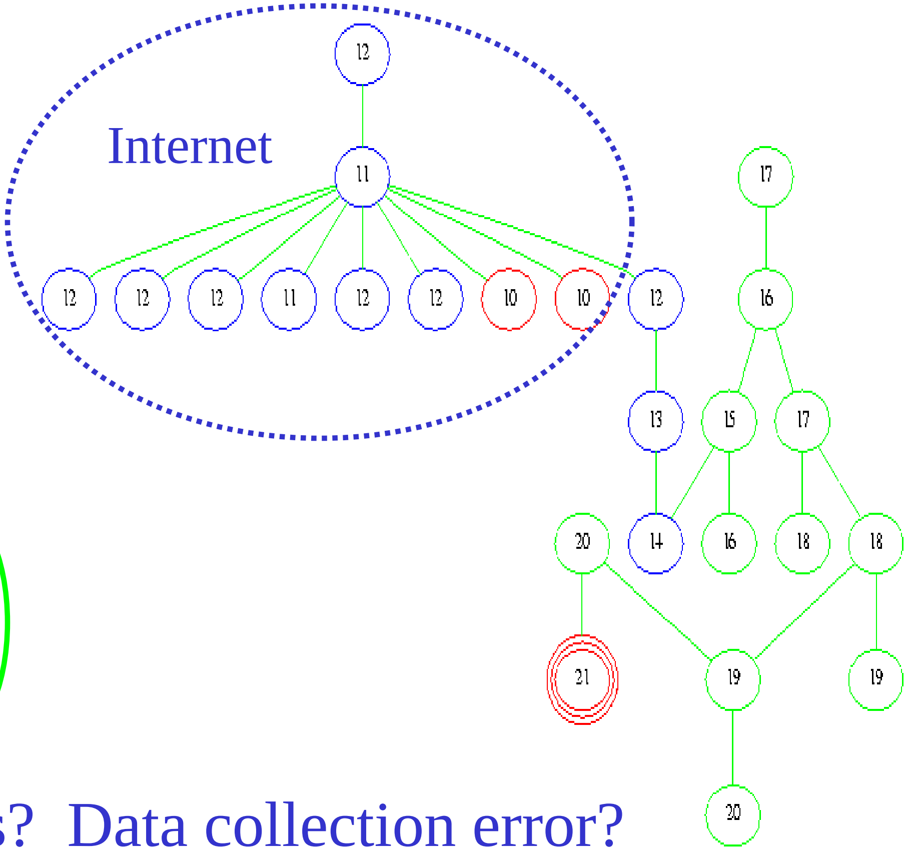
- Topologically important nodes: very well connected.
- Conjecture: These are "core" routers in the Internet..

"Poor" Nodes ?



“Poor” Nodes ?





Who and what are these nodes? Data collection error? Poorly connected countries? Other?

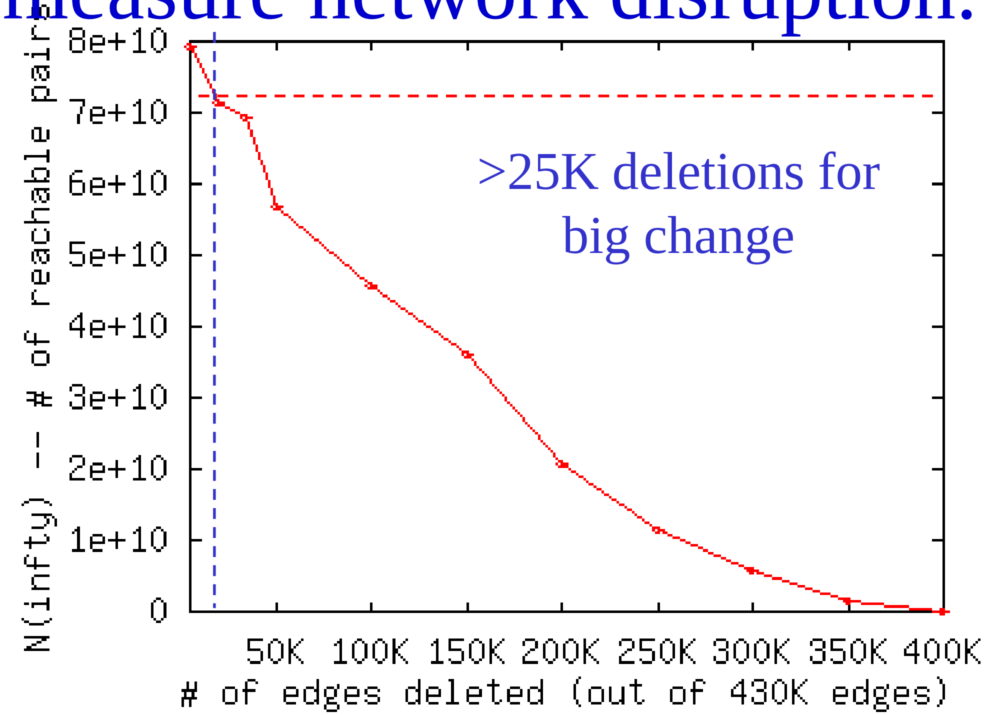
(Approx.) neighborhood function

- Useful for estimating the diameter of a graph;
- the ``effective radius'' of a node (distance to 90%-tile of the other nodes)
- the connectivity under failures
- quick checks for (dis-)similarity between two graphs

Link Failures

Experiment: Pick an edge at random, delete it and measure network disruption.

#pairs



#deleted edges

Internet very resilient to link failures

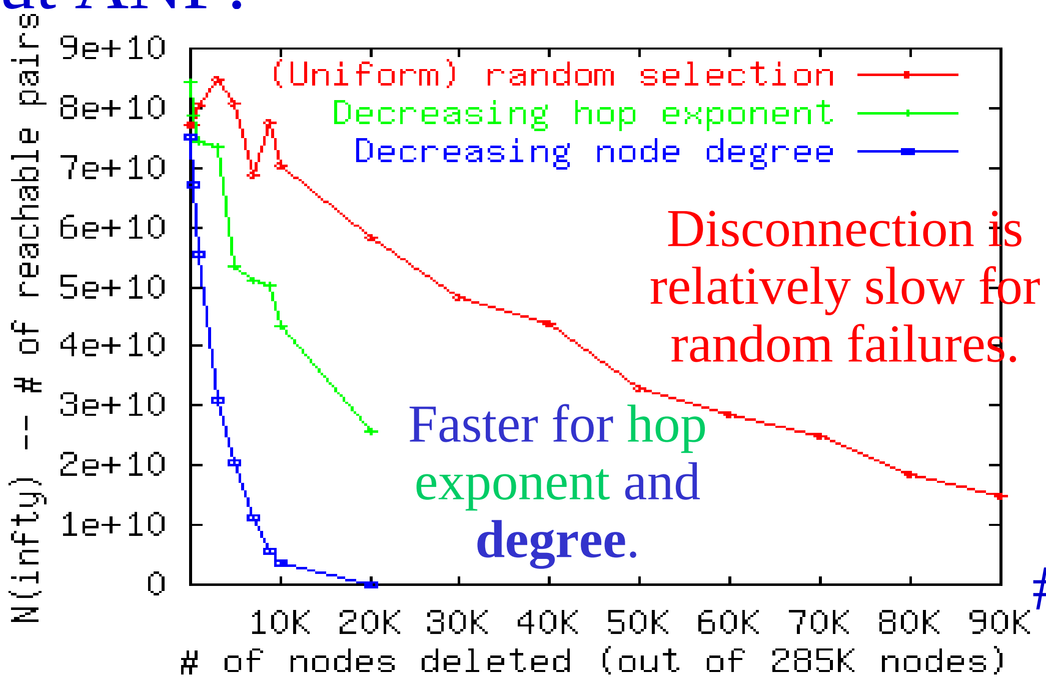
Effect of node deletions

- Robust to random failures, focussed failures are a problem
- What is best way to break connectivity:
  - delete highest degree first? or
  - delete highest hop-exponent (~smalles radius) first?

Effect of node deletions

- Robust to random failures, focussed failures are a problem
- ALL these runs would take >100x times longer without ANF!

#pairs



#deleted nodes

Outline

- Introduction - motivation
- Patterns & Power laws
- Scalability & Fast algorithms
- **Fractals, graphs and power laws**
- Conclusions

Why power laws appear at all?

Q: Why do they appear so often? (Pareto, Lotka, Gutenberg-Richter, Sirbu, ...)

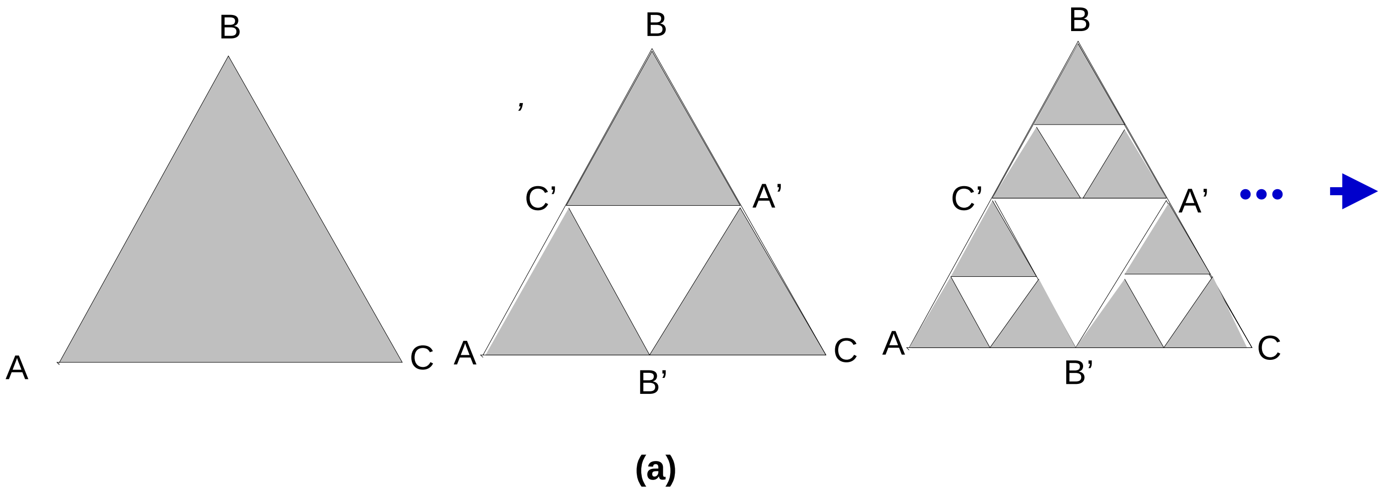
Why power laws?

Q: Why do they appear so often? (Pareto, Lotka, Gutenberg-Richter, Sirbu, ...)

A: One possible explanation: self-similarity / recursion / fractals – in detail:

What is a fractal?

= **self-similar** point set, e.g., Sierpinski triangle:



zero area;

infinite length!

Q: What is its dimensionality??

A: log3 / log2 = 1.58 (!?!)

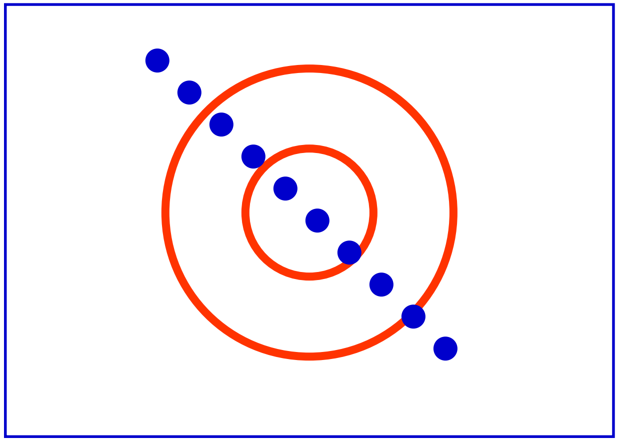
Intrinsic (‘fractal’) dimension

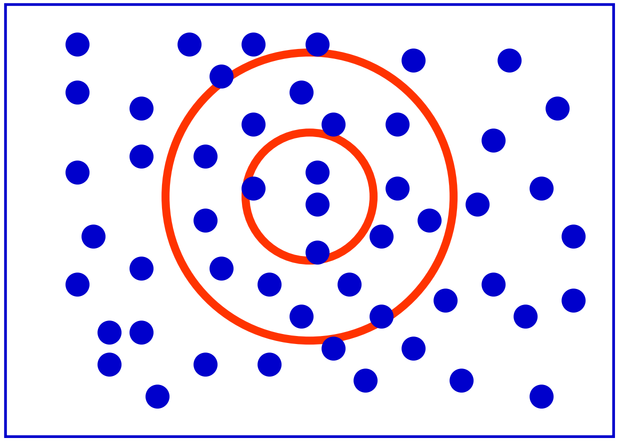
- Q: fractal dimension of a line?
- A: nn (<= r) ~ r^1

(‘power law’: y=x^a)

- Q: fd of a plane?
- A: nn (<= r) ~ r^2

fd== slope of (log(nn) vs.. log(r))



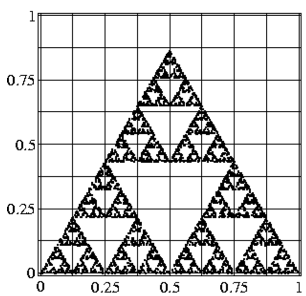


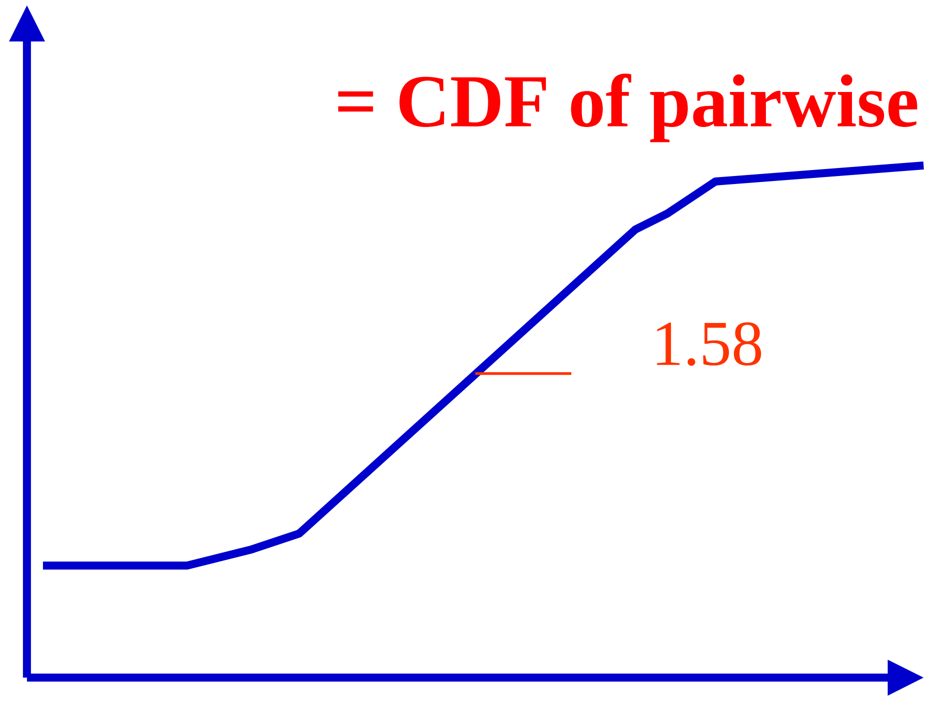
Sierpinsky triangle

== 'correlation integral'

= CDF of pairwise distances

log(#pairs within <=r)

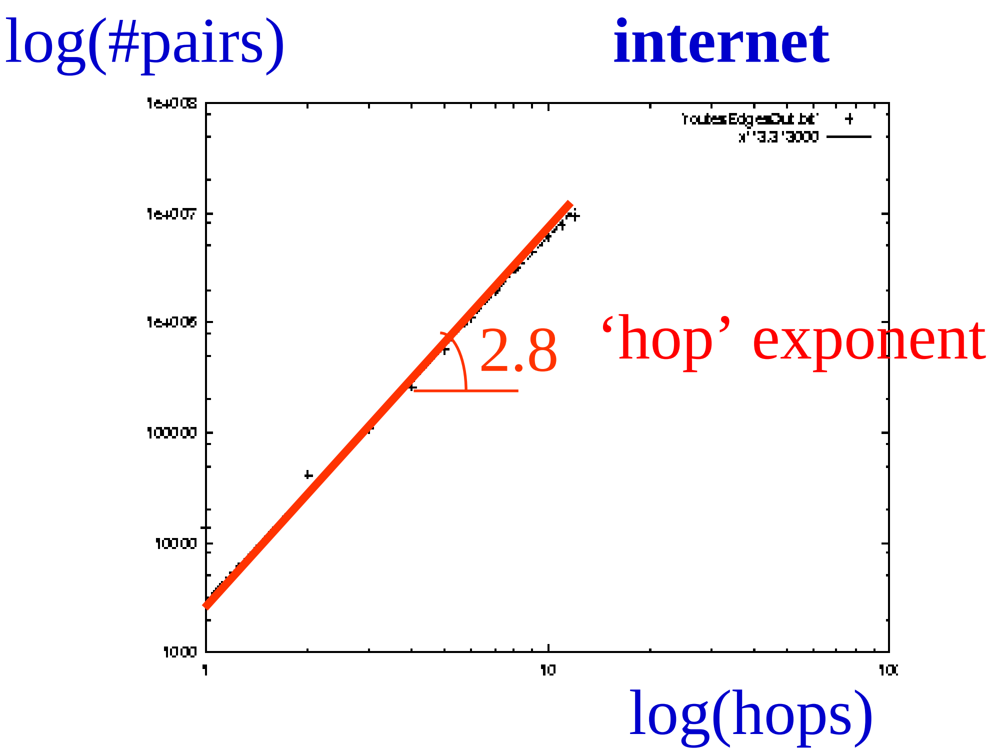




log(r)

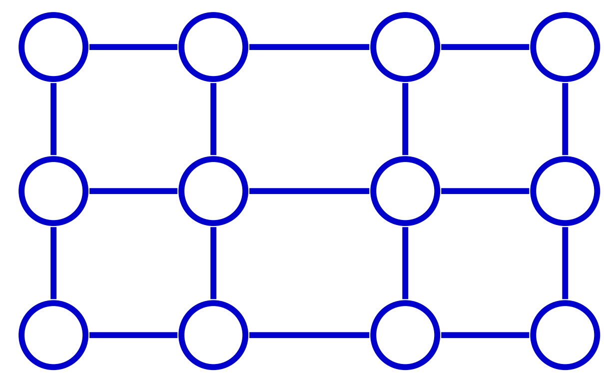
Solution#1'': Hop Exponent *H*

- A3: neighborhood function *N(h)* = number of pairs within *h* hops or less - power law, too!



Hop exp. = 1

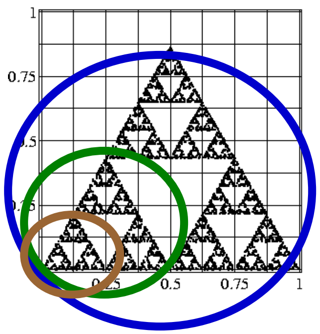
Hop exp. = 2

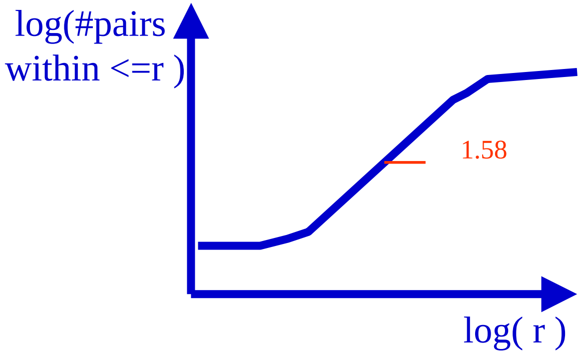


Observations: Fractals <-> power laws

Closely related:

- fractals <=>
- self-similarity <=>
- scale-free <=>
- power laws ($y= x^a$)
- (vs $y=e^{-ax}$ or $y=x^a+b$)





Fractals in nature

- Q: How often do they appear in practice?
- A: extremely often!
  - coastlines (~1.2)
  - mammalian brain surface (~2.6)
  - bark of trees (~2.1)
  - ...

[See Schroeder: "Fractals, Chaos & Power laws"]

Fractals – discussion

- Also related to fractals/self-similarity:
  - phase transitions / renormalization / Ising spins
  - cellular automata
  - self-organized criticality (SOC) [Bak]
  - long-range dependency / heavy tailed distr. in network traffic [Leland+]

To iterate is human; to recurse is divine

Conclusions

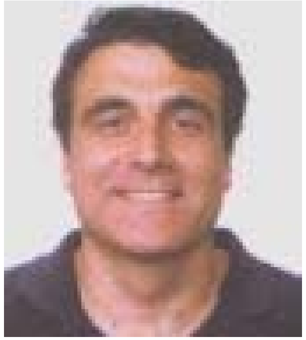
- Many real graphs/networks follow ‘power laws’ (~ fractals ~ self-similarity)
  - and continue that over time
- We need fast, scalable algorithms for large graphs, like ‘ANF’
- Cross-disciplinarity: pays off (DB + Theory + Networks + Physics + …)

Thank you!

Contact info:

christos@cs.cmu.edu

www.cs.cmu.edu/~christos



- Code for fractal dimension: on the web
- Network data:
  - CAIDA caida.org ;
  - NLANR nlanr.net