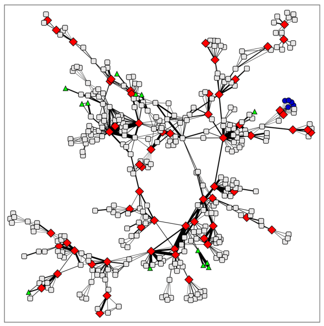
Discovering Roles and Anomalies in Graphs: Theory and Applications

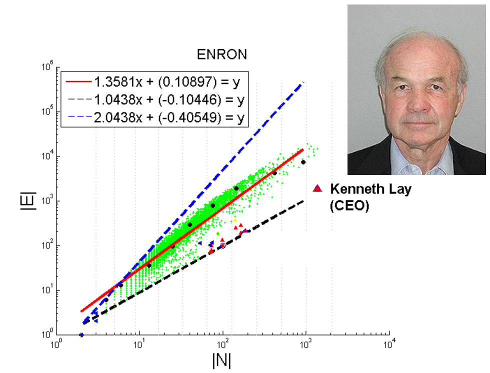
Part 2: patterns, anomalies and applications

*Tina Eliassi-Rad (*Rutgers*)*

*Christos Faloutsos (*CMU*)*

OVERVIEW - high level:



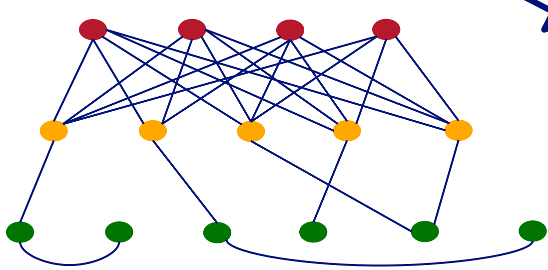


Features

Roles

Anomalies

= rare roles



Patterns

Resource:

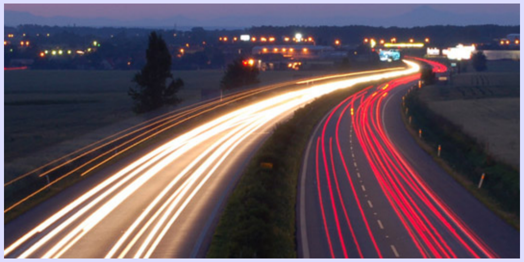
Open source system for mining huge graphs:

PEGASUS project (PEta GrAph mining System)

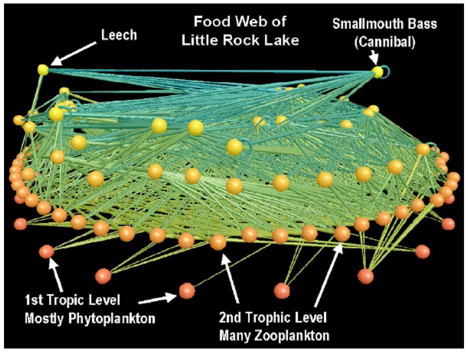
- www.cs.cmu.edu/~pegasus
- code and papers

Roadmap

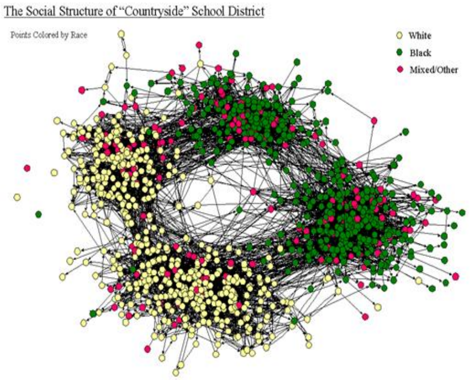
- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions



Graphs - why should we care?



Food Web
[Martinez '91]

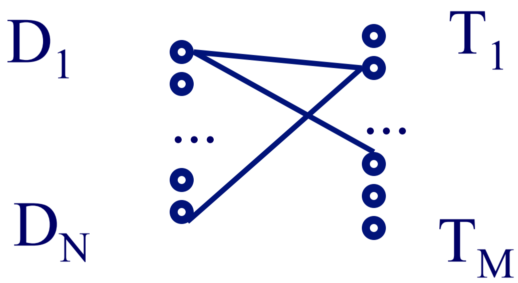


Friendship Network
[Moody '01]

Internet Map
[lumeta.com]

Graphs - why should we care?

- IR: bi-partite graphs (doc-terms)



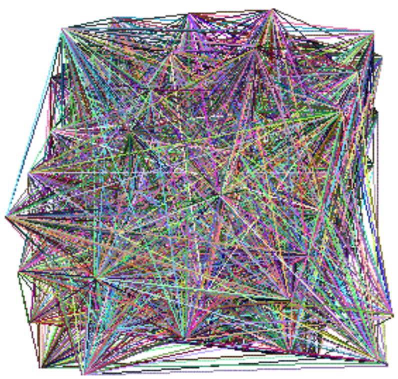
- web: hyper-text graph

- ... and more:

Graphs - why should we care?

- 'viral' marketing
- web-log ('blog') news propagation
- computer network security: email/IP traffic and anomaly detection
- ....

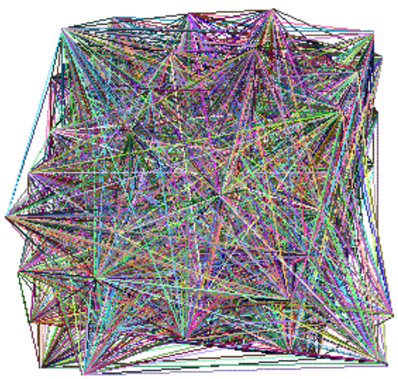
Problem #1 - network and graph mining



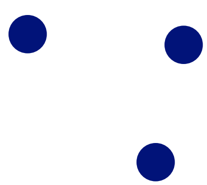
- What does the Internet look like?
- What does FaceBook look like?

- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?

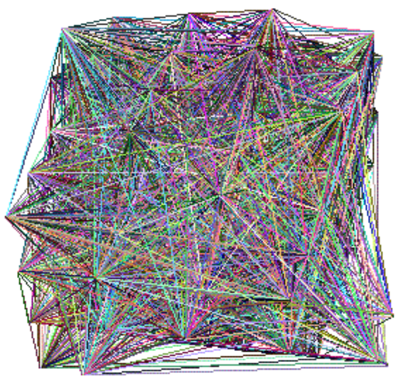
Problem #1 - network and graph mining



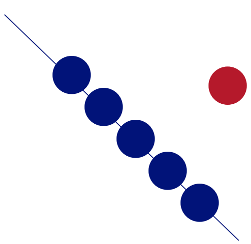
- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**



Problem #1 - network and graph mining



- What does the Internet look like?
- What does FaceBook look like?
- What is 'normal'/'abnormal'?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**
  - **Large** datasets reveal patterns/anomalies that may be invisible otherwise…



Graph mining

- Are real graphs random?

Laws and patterns

- Are real graphs random?
- A: NO!!
  – Diameter
  – in- and out- degree distributions
  – other (surprising) patterns

- So, let's look at the data

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

Carnegie Mellon

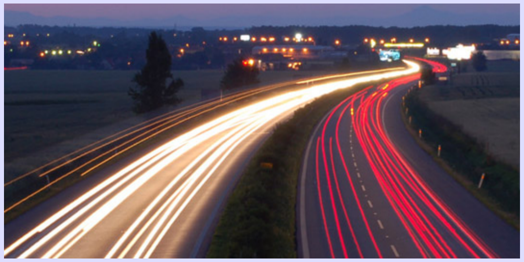
Real Graph Patterns

	unweighted	weighted
static	**P01**. Power-law degree distribution [Faloutsos et. al.`99, Kleinberg et. al.`99, Chakrabarti et. al. `04, Newman`04] **P02**. Triangle Power Law [Tsourakakis `08] **P03**. Eigenvalue Power Law [Siganos et. al. `03] **P04**. Community structure [Flake et. al.`02, Girvan and Newman `02] **P05**. Clique Power Laws [Du et. al. '09]	**P12**. Snapshot Power Law [McGlohon et. al. `08]
dynamic	**P06**. Densification Power Law [Leskovec et. al.`05] **P07**. Small and shrinking diameter [Albert and Barabási `99, Leskovec et. al. '05, McGlohon et. al. '08] **P08**. Gelling point [McGlohon et. al. `08] **P09**. Constant size $2^{nd}$ and $3^{rd}$ connected components [McGlohon et. al. `08] **P10**. Principal Eigenvalue Power Law [Akoglu et. al. `08] **P11**. Bursty/self-similar edge/weight additions [Gomez and Santonja `98, Gribble et. al. `98, Crovella and Bestavros `99, McGlohon et .al. `08]	**P13**. Weight Power Law [McGlohon et. al. `08] **P14**. Skewed call duration distributions [Vaz de Melo et. al. `10]

RTG: A Recursive Realistic Graph Generator using Random Typing

Leman Akoglu and Christos Faloutsos. *ECML PKDD*'09.

Roadmap

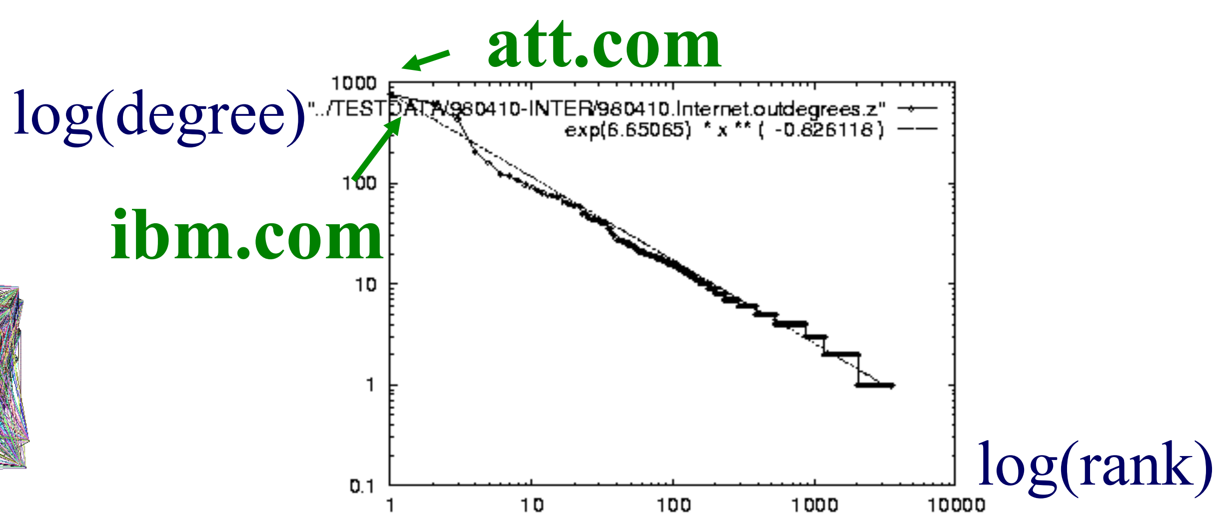


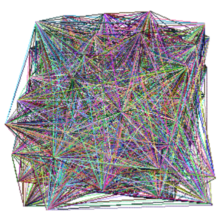
- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

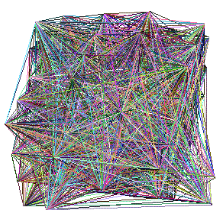


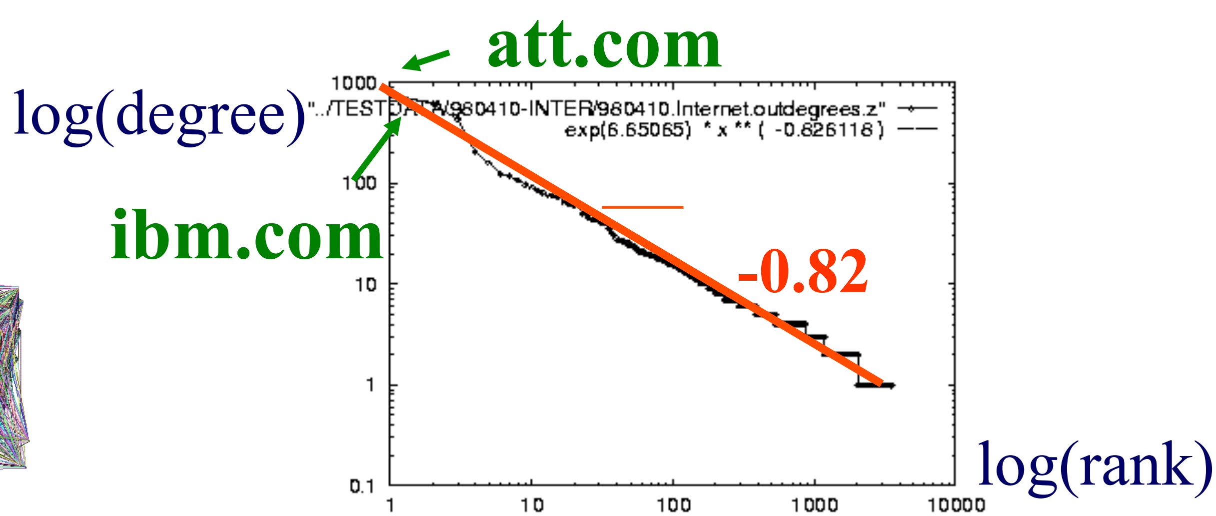


Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

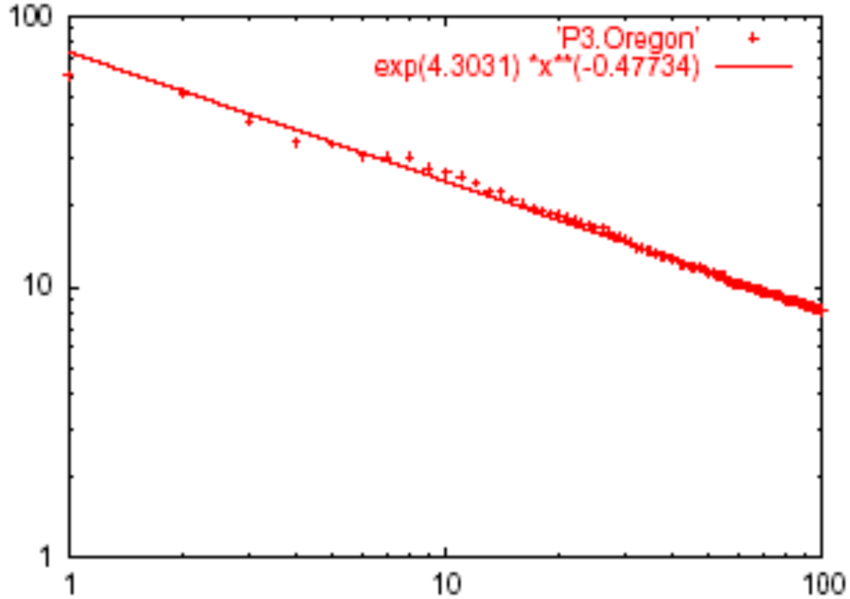
internet domains





Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

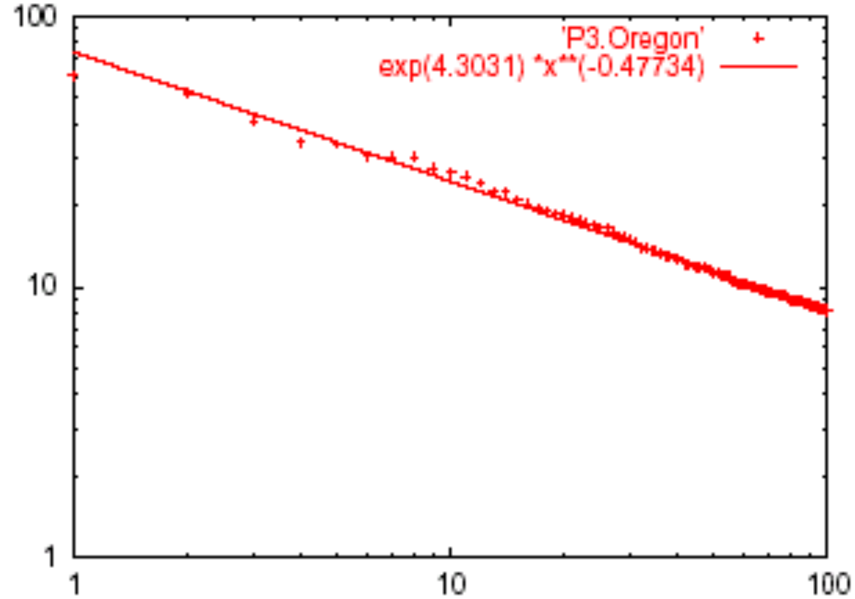
E = -0.48

May 2001

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

Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

E = -0.48

May 2001

- [Mihail, Papadimitriou ’02]: slope is ½ of rank exponent

But:

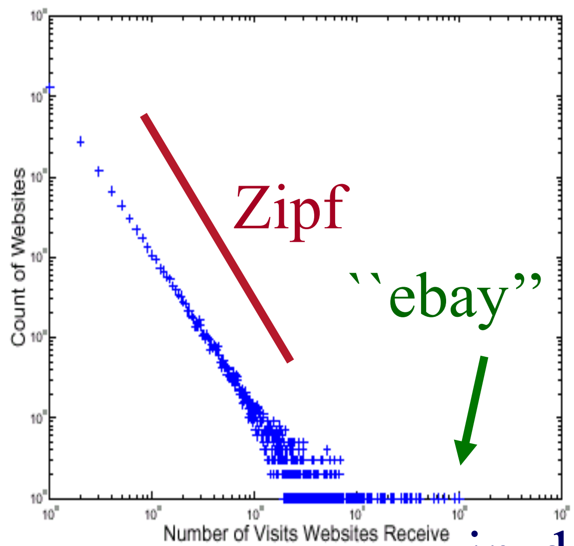
How about graphs from other domains?

More power laws:

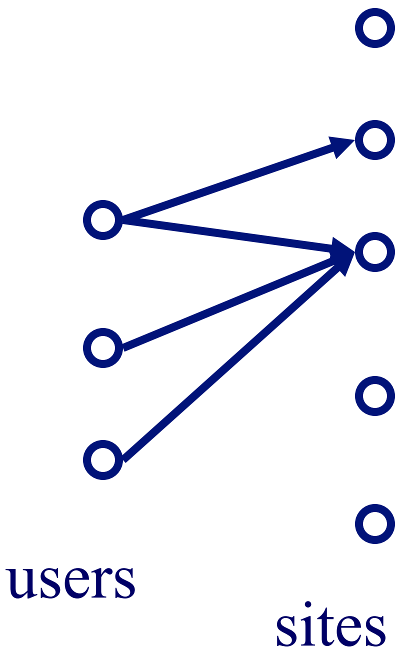
- web hit counts [w/ A. Montgomery]

Web Site Traffic

Count
(log scale)



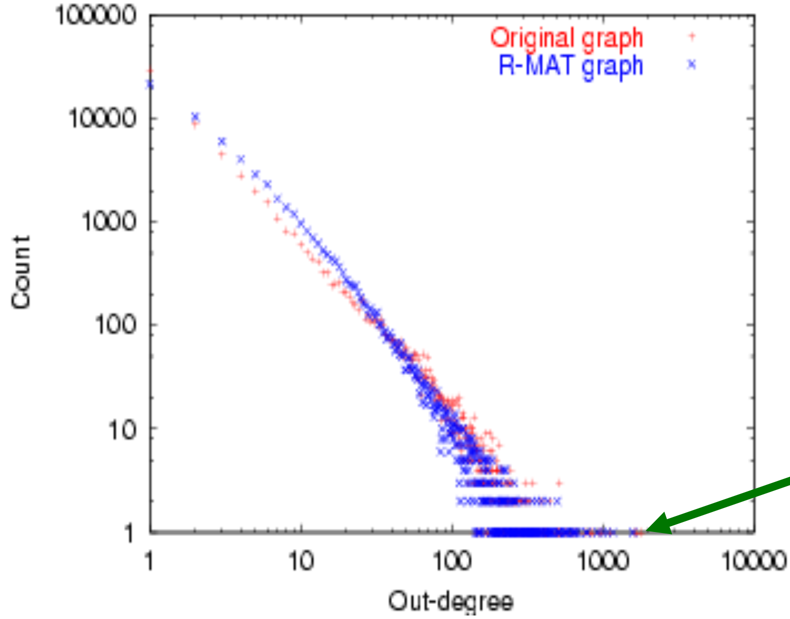
in-degree (log scale)



epinions.com

- who-trusts-whom [Richardson + Domingos, KDD 2001]

count



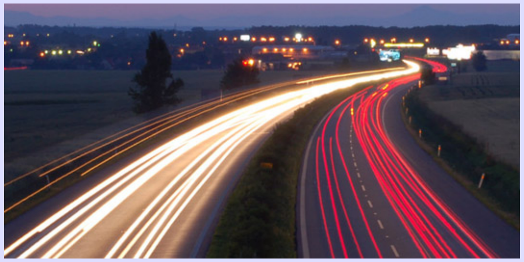
trusts-2000-people user

(out) degree

And numerous more

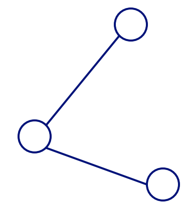
- # of sexual contacts
- Income [Pareto] –'80-20 distribution'
- Duration of downloads [Bestavros+]
- Duration of UNIX jobs ('mice and elephants')
- Size of files of a user
- …
- 'Black swans'

Roadmap



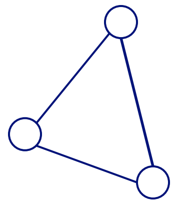
- Patterns in graphs
  - overview
  - Static graphs
    - S1: Degree, S2: eigenvalues
    - S3-4: Triangles, S5: cliques
    - Radius plot
    - Other observations (‘eigenSpokes’)
  - Weighted graphs
  - Time-evolving graphs

Solution# S.3: Triangle ‘Laws’



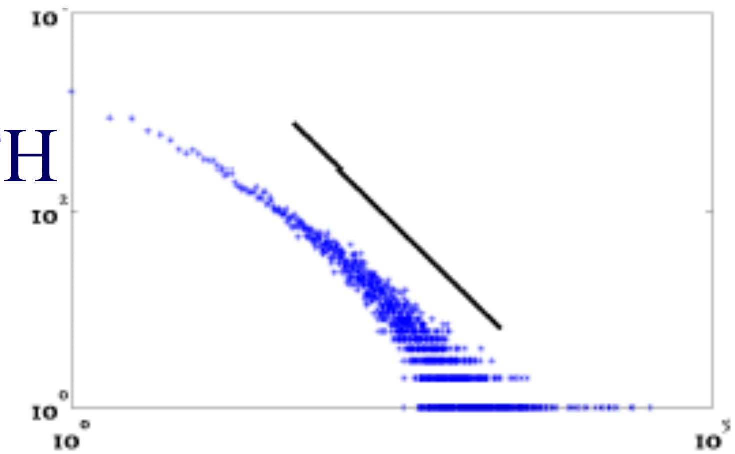
- Real social networks have a lot of triangles

Solution# S.3: Triangle 'Laws'

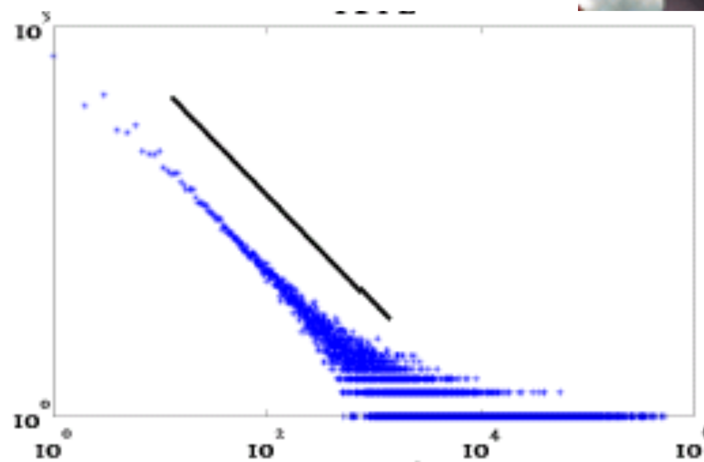


- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

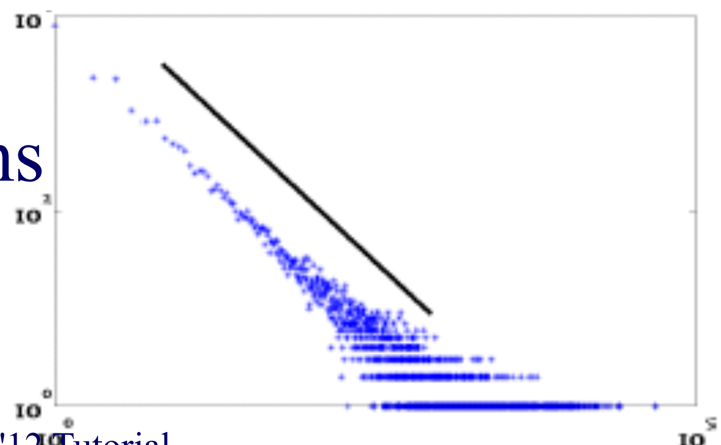
Triangle Law: #S.3
[Tsourakakis ICDM 2008]



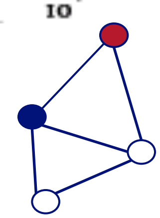
HEP-TH



ASN



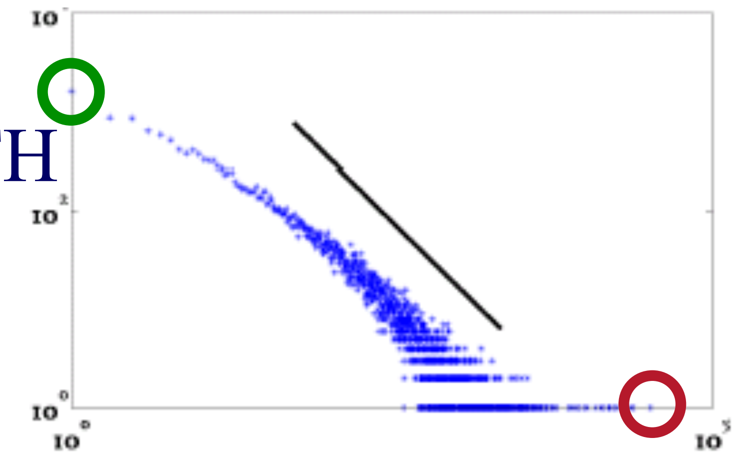
Epinions



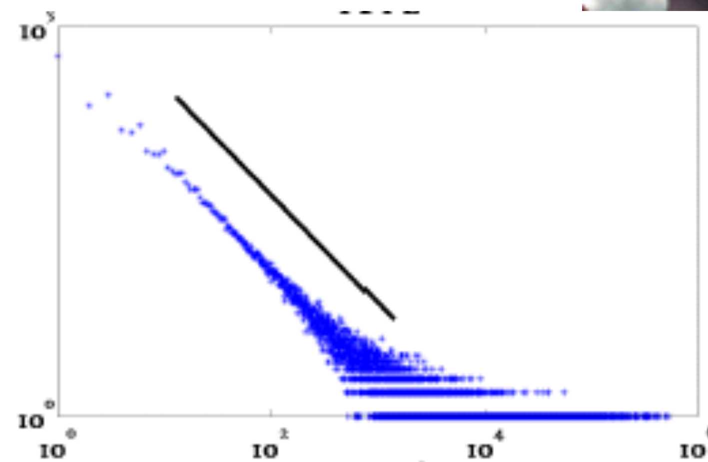
X-axis: # of participating triangles
Y: count (~ pdf)

Triangle Law: #S.3
[Tsourakakis ICDM 2008]

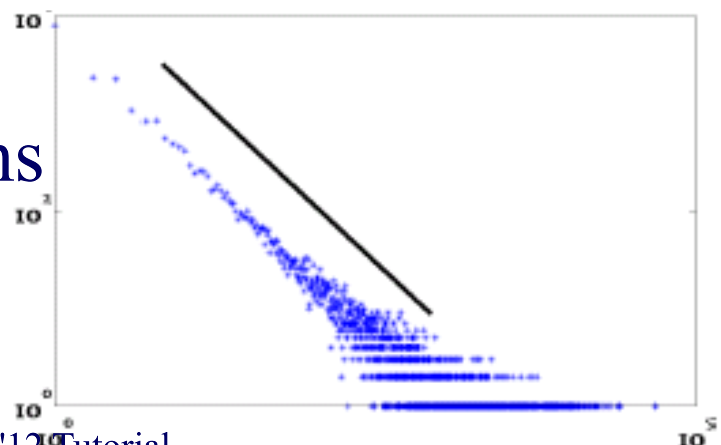
HEP-TH

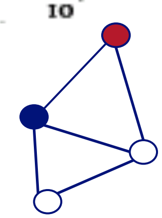


ASN



Epinions



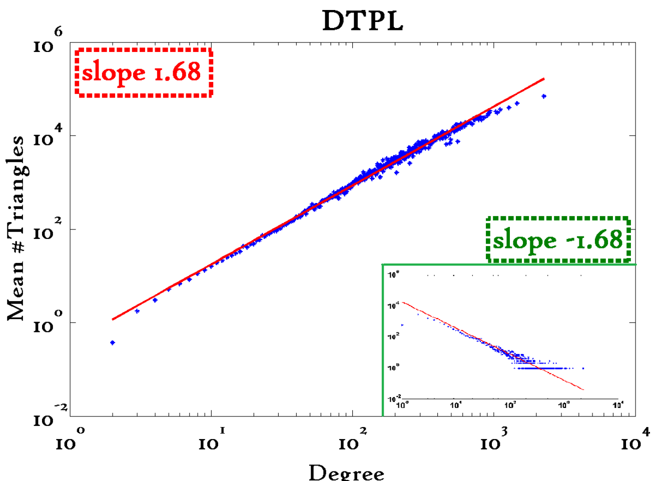


X-axis: # of participating triangles
Y: count (~ pdf)

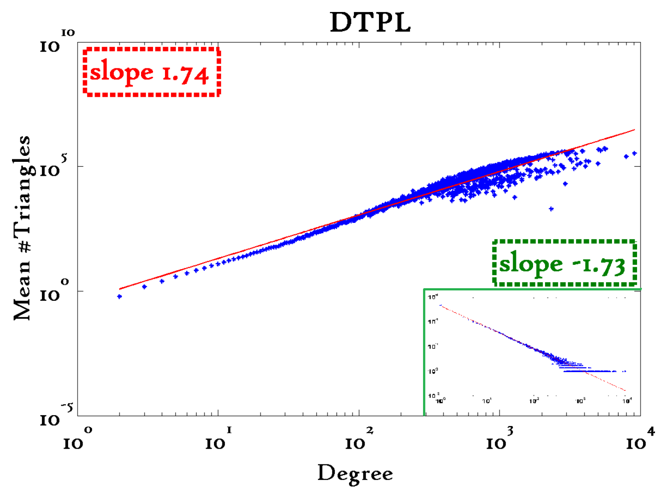
Triangle Law: #S.4

[Tsourakakis ICDM 2008]

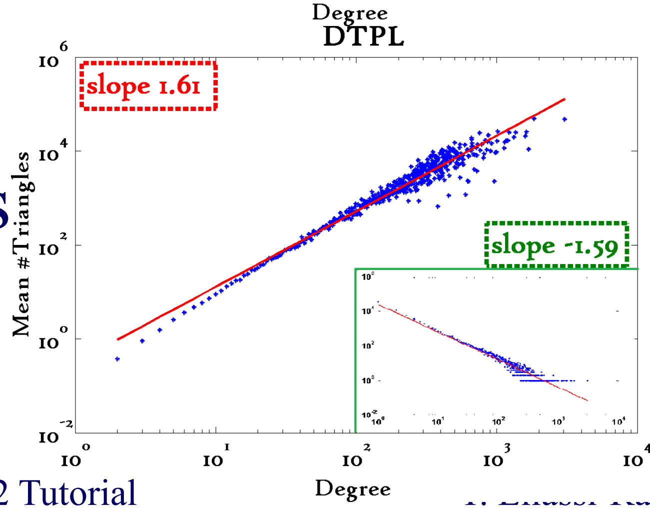
Reuters



SN



Epinions



X-axis: degree

Y-axis: mean # triangles

n friends -> ~$n^{1.6}$ triangles

details

Triangle Law: Computations [Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos)
Q: Can we do that quickly?

details

Triangle Law: Computations [Tsourakakis ICDM 2008]

But: triangles are expensive to compute
(3-way join; several approx. algos)

Q: Can we do that quickly?

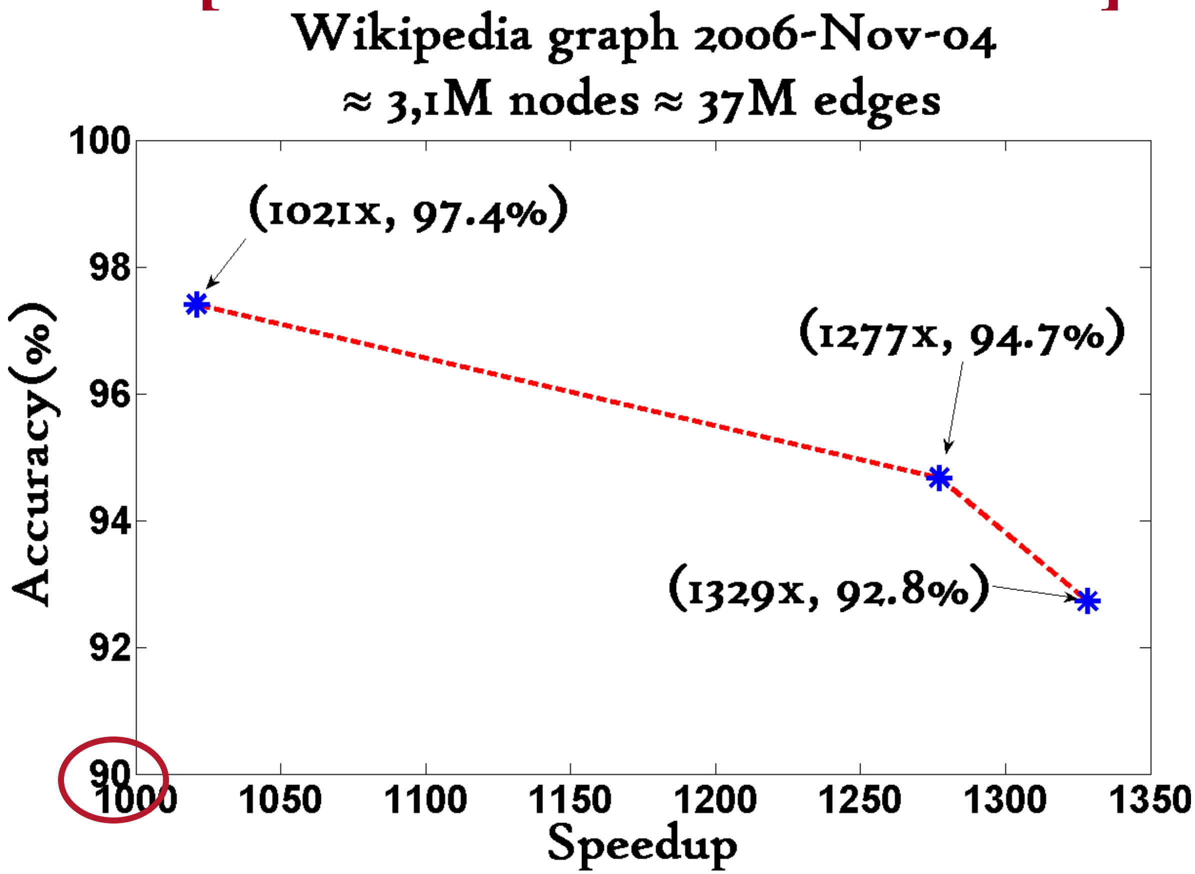
A: Yes!

#triangles = 1/6 Sum (λ_i^3)

(and, because of skewness (S2) ,
we only need the top few eigenvalues!

details

Triangle Law: Computations
[Tsourakakis ICDM 2008]



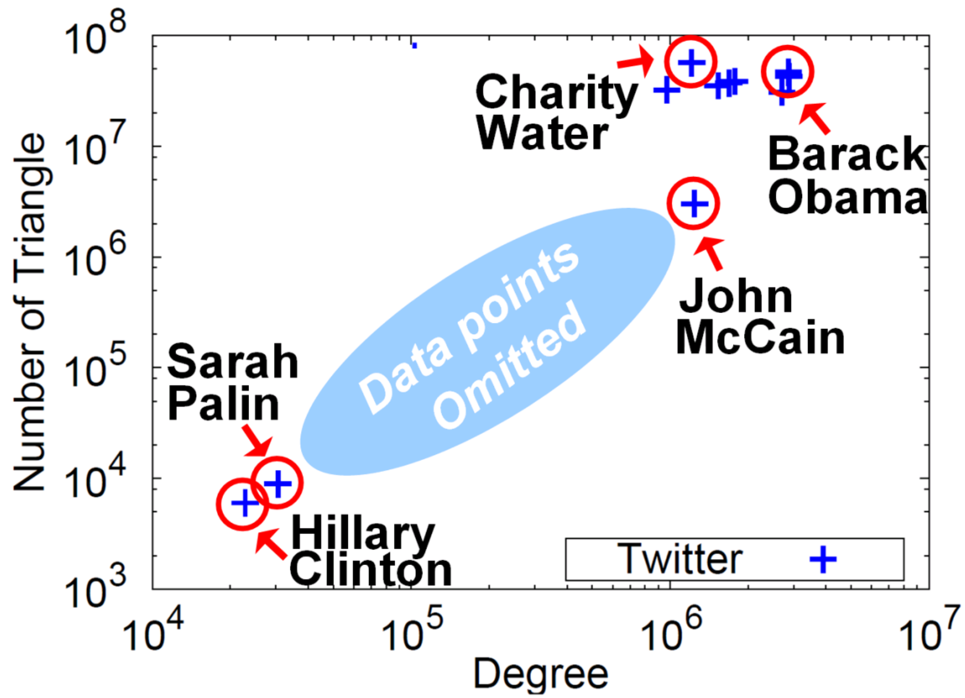
1000x+ speed-up, >90% accuracy

Triangle counting for large graphs?

Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD'11]

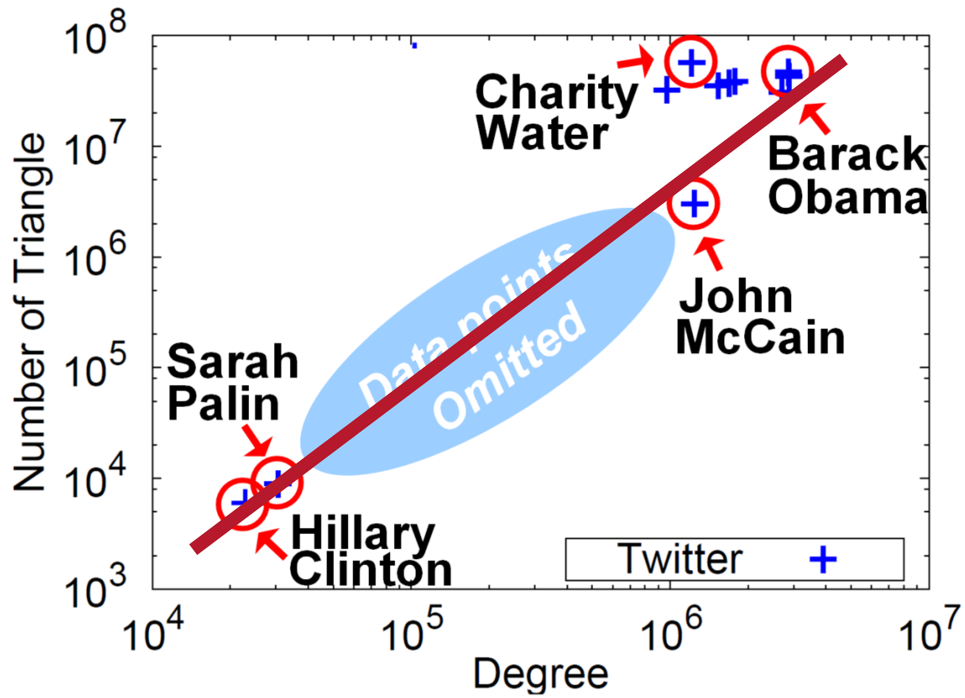
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD'11]

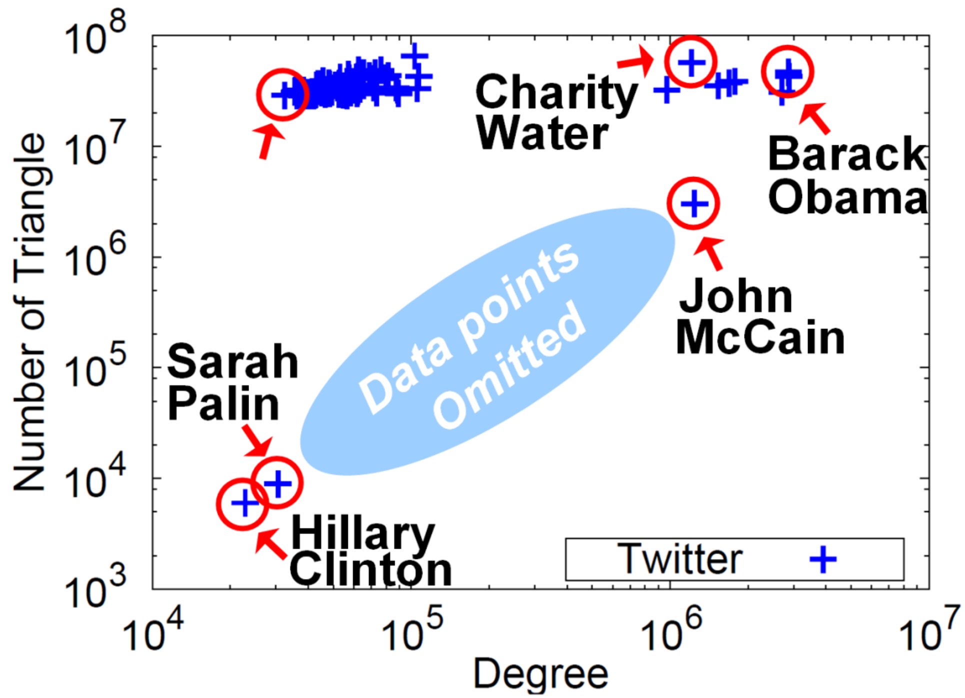
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD'11]

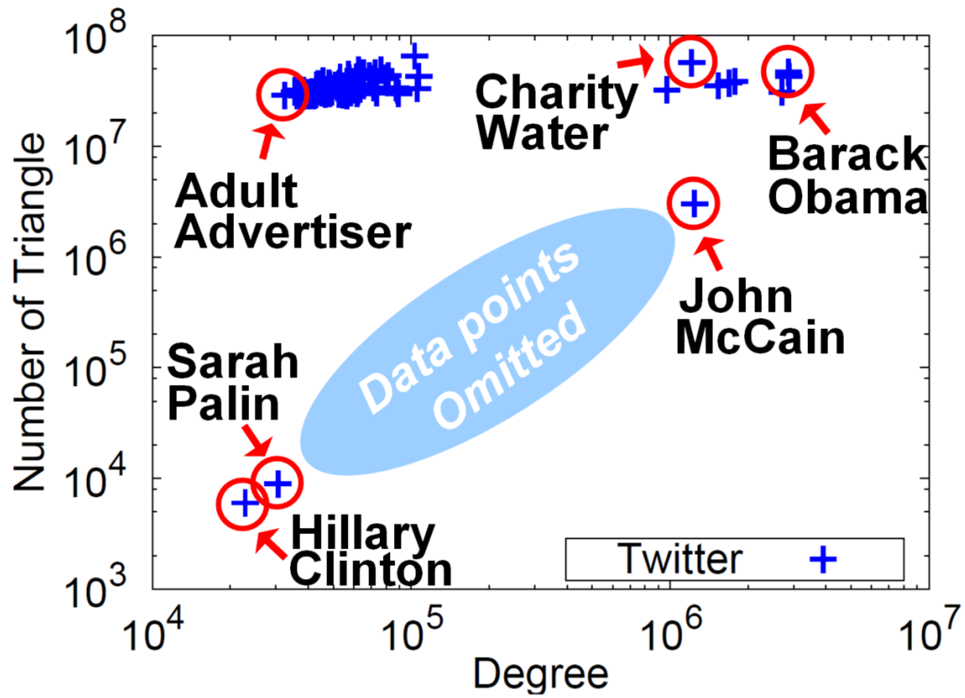
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD'11]

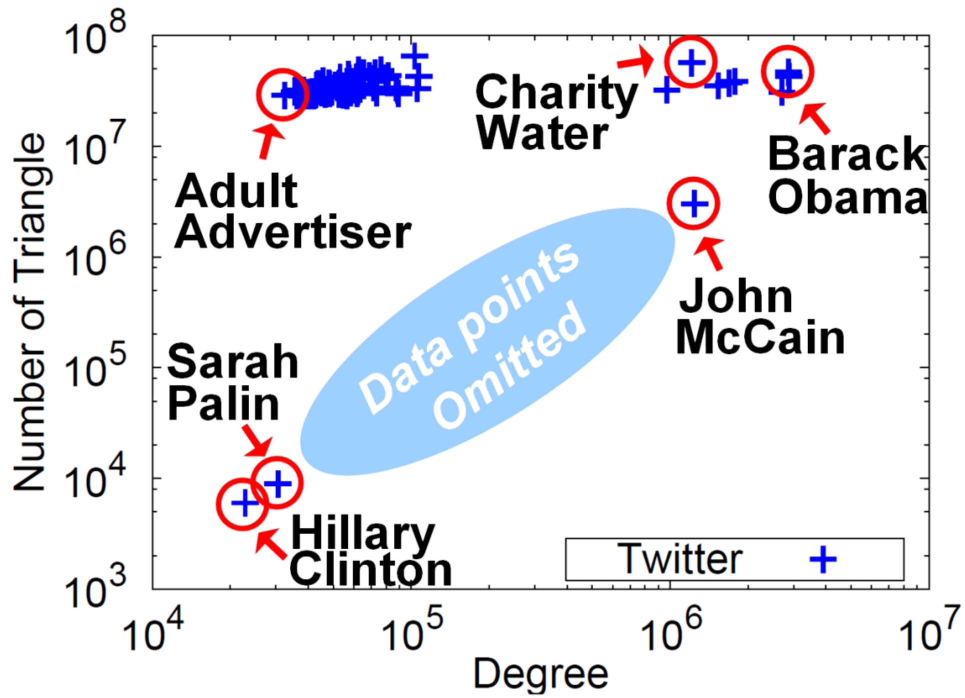
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

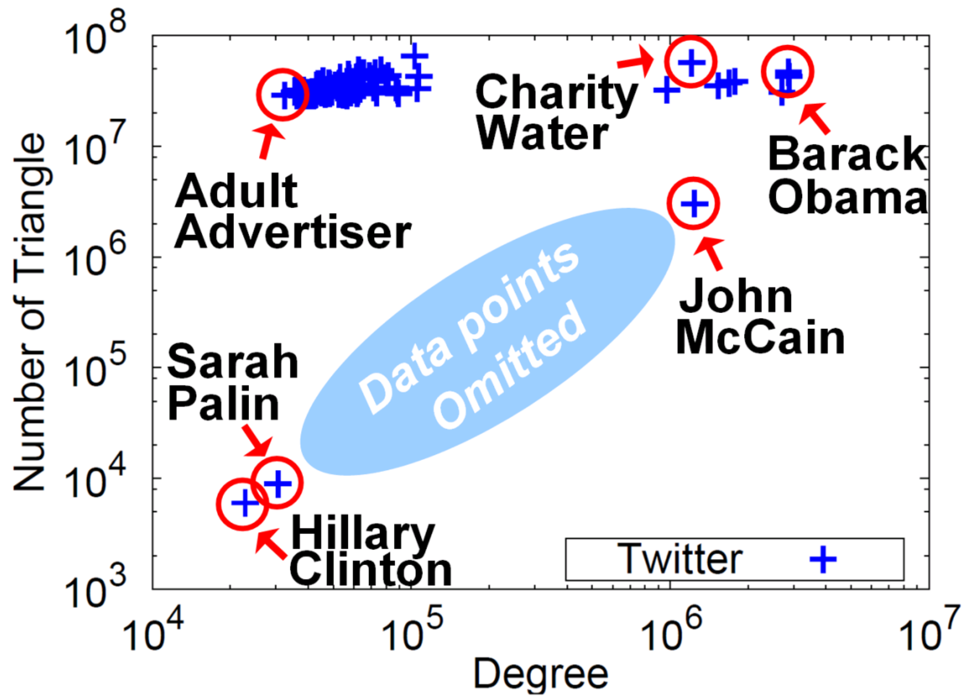
[U Kang, Brendan Meeder, +, PAKDD'11]

Triangle counting for large graphs?



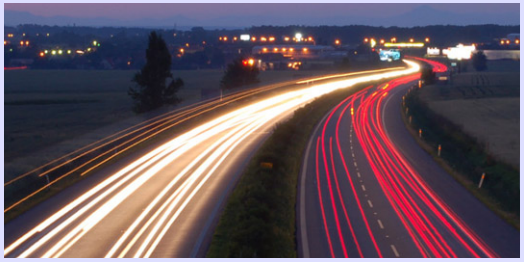
Q: How to compute # triangles in B-node graph? (O(d_{max} ** 2))?

Triangle counting for large graphs?



Q: How to compute # triangles in B-node graph? (O(d_{max} ** 2))? **A: cubes of eigvals**

Roadmap



- Patterns in graphs
  - overview
  - Static graphs
    - S1: Degree, S2: eigenvalues
    - S3-4: Triangles, S5: cliques
    - Radius plot
    - Other observations ('eigenSpokes')
  - Weighted graphs
  - Time-evolving graphs

How about cliques?

Large Human Communication Networks

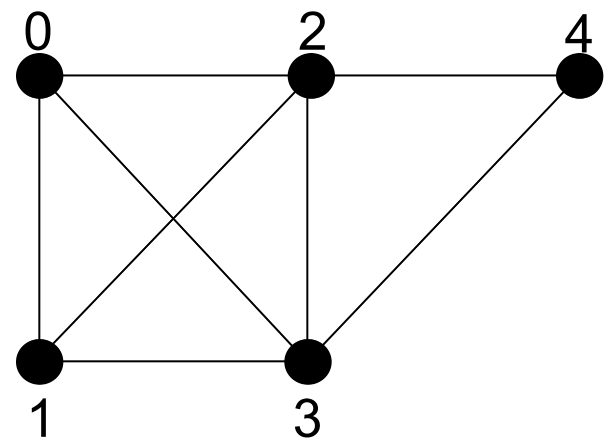
Patterns and a Utility-Driven Generator

Nan Du, Christos Faloutsos, Bai Wang, Leman Akoglu

KDD 2009

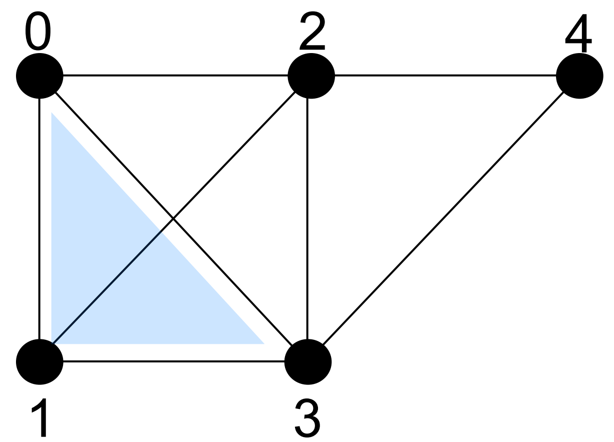
Cliques

- Clique is a complete subgraph.
- If a clique can not be contained by any larger clique, it is called the maximal clique.



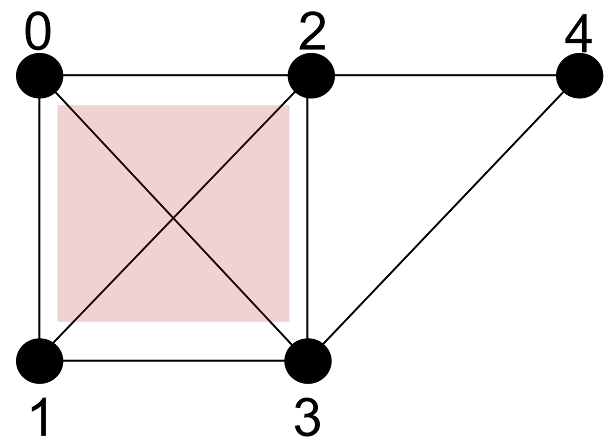
Clique

- Clique is a complete subgraph.
- If a clique can not be contained by any larger clique, it is called the maximal clique.



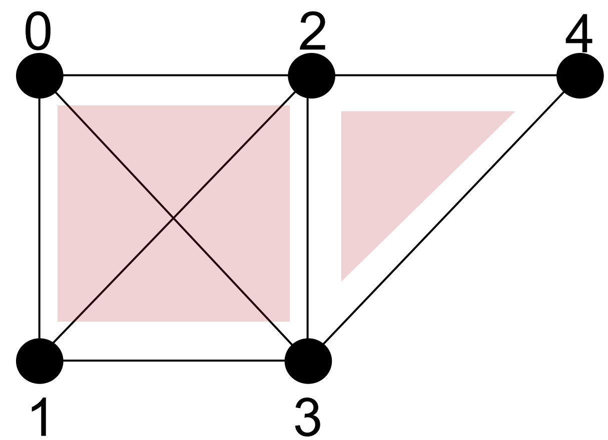
Clique

- Clique is a complete subgraph.
- If a clique can not be contained by any larger clique, it is called the maximal clique.



Clique

- Clique is a complete subgraph.
- If a clique can not be contained by any larger clique, it is called the maximal clique.
- {0,1,2}, {0,1,3}, {1,2,3} {2,3,4}, {0,1,2,3} are cliques;
- {**0,1,2,3**} and {**2,3,4**} are the maximal cliques.



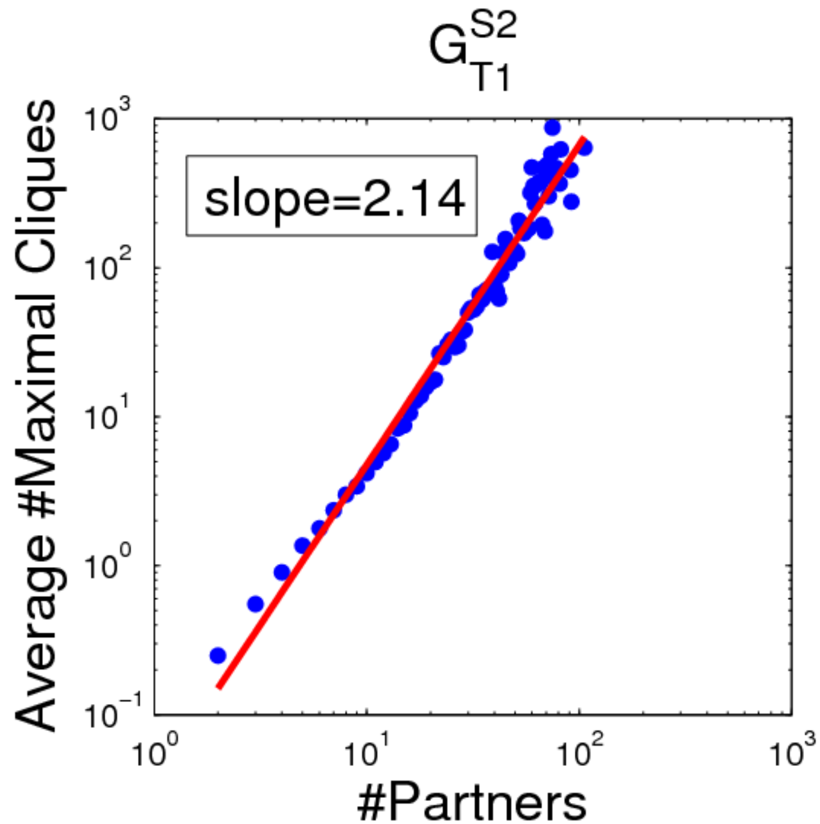
S5: Clique-Degree Power-Law

- Power law:

$$C_{avg}^{d_i} \propto d_i^{\alpha}$$

maximal cliques of node i

degree of node i

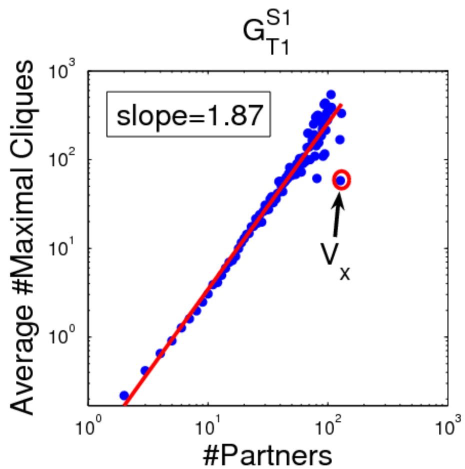


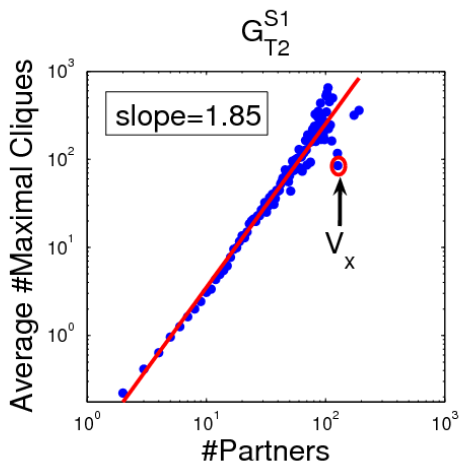
α is the power law exponent
$\alpha \in [1.8, 2.2]$ for S1~S3

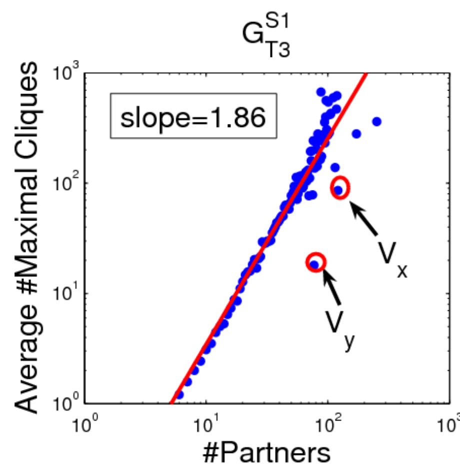
More friends, even more social circles !

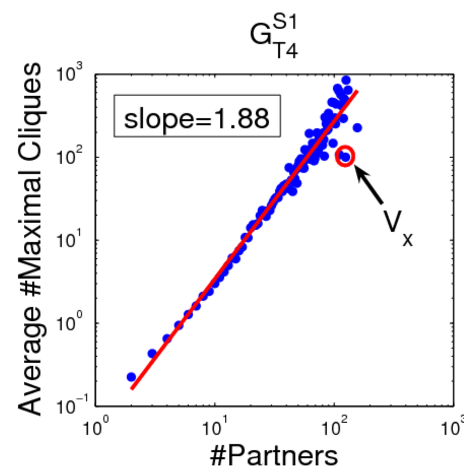
S5: Clique-Degree Power-Law

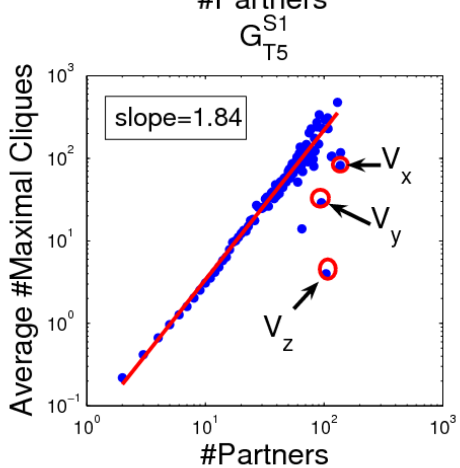
- Outlier Detection





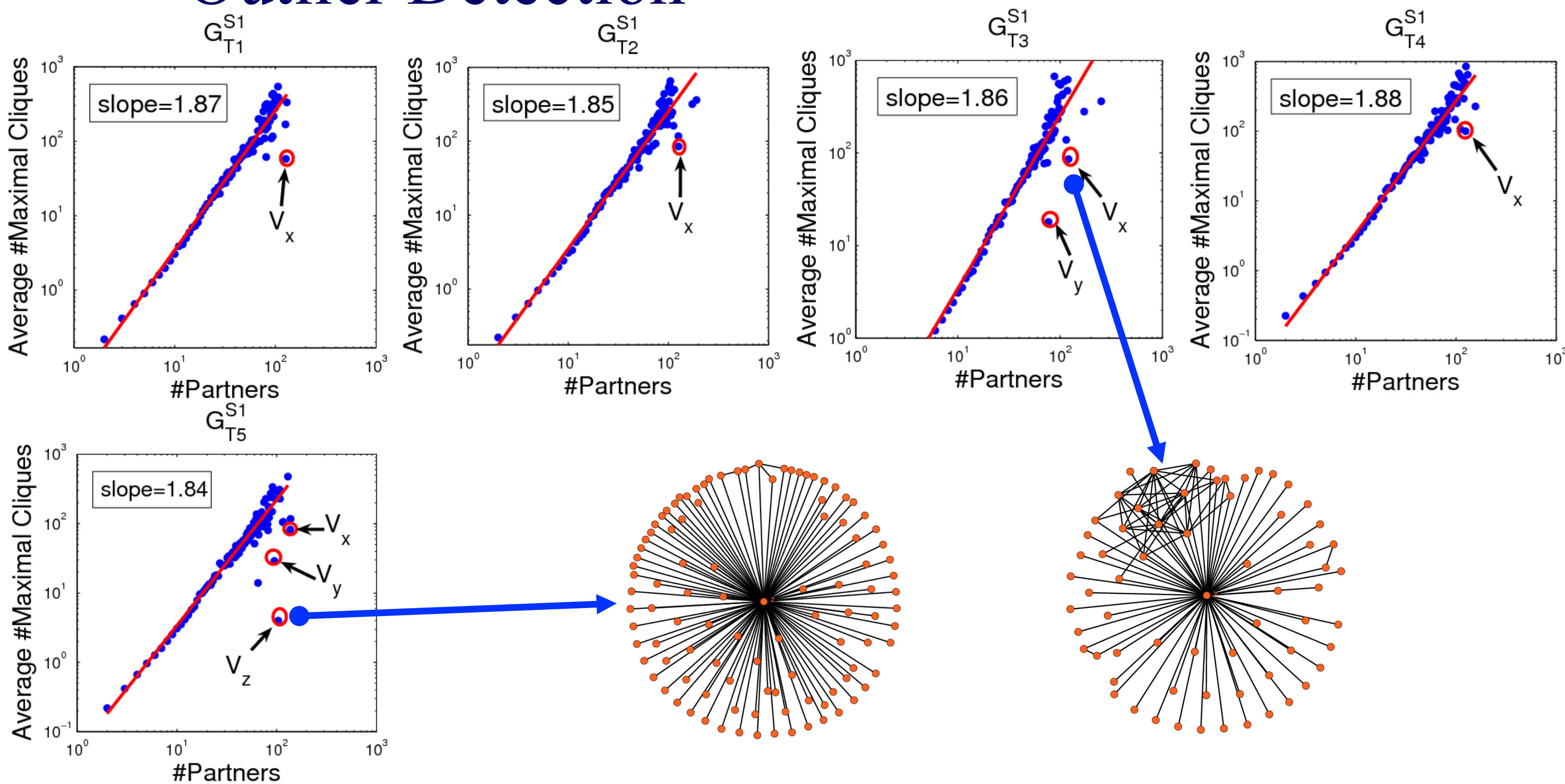




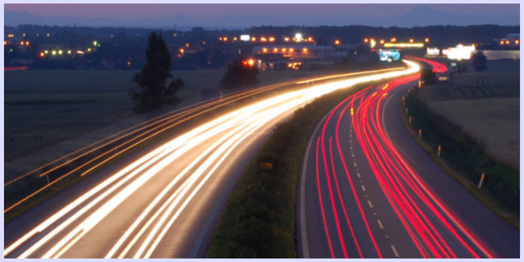


S5: Clique-Degree Power-Law

- Outlier Detection



Roadmap

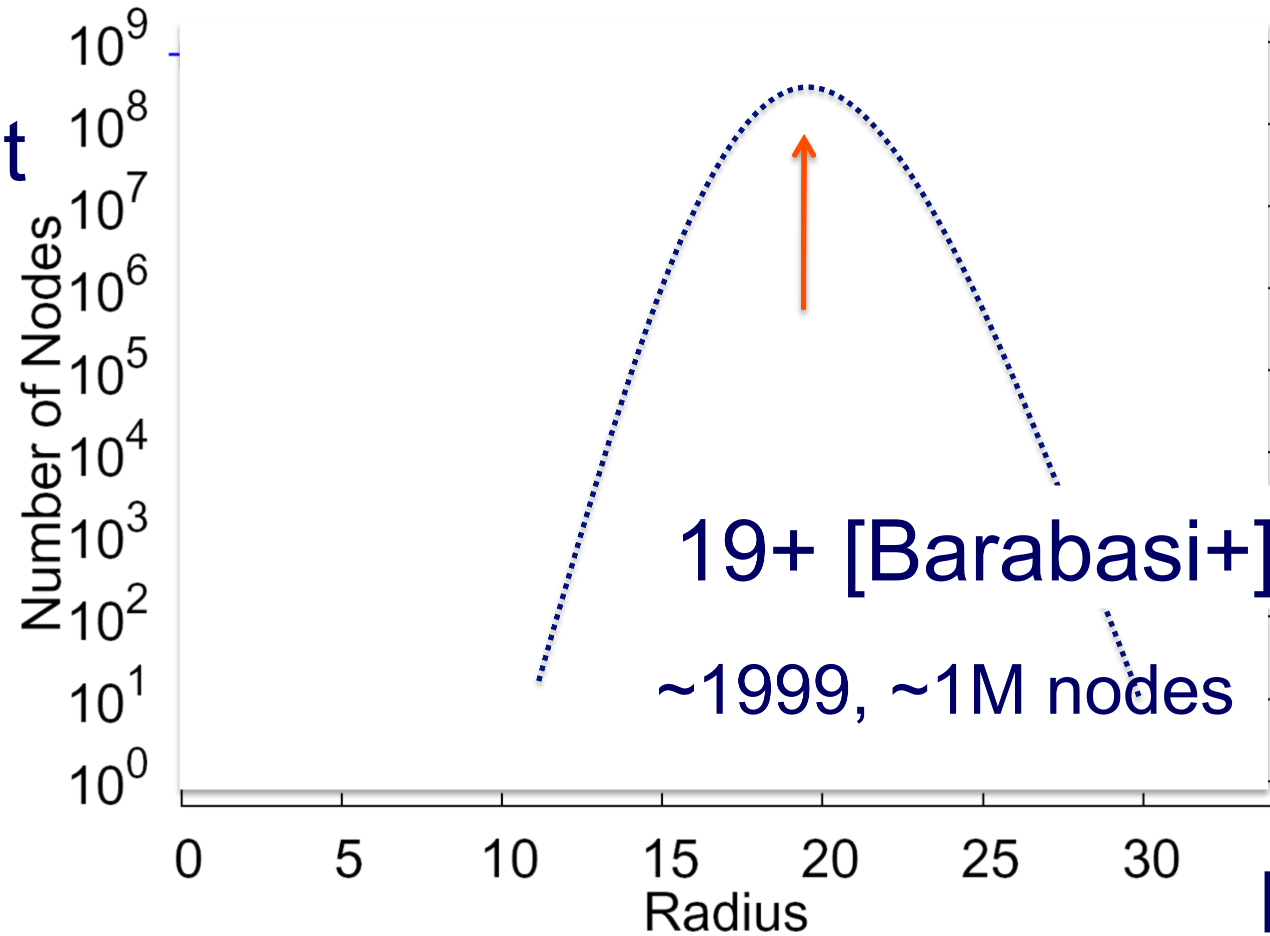


- Patterns in graphs
  - overview
  - Static graphs
    - S1: Degree, S2: eigenvalues
    - S3-4: Triangles, S5: cliques
    - Radius plot
    - Other observations (‘eigenSpokes’)
  - Weighted graphs
  - Time-evolving graphs

HADI for diameter estimation

- *Radius Plots for Mining Tera-byte Scale Graphs* **U Kang**, Charalampos Tsourakakis, Ana Paula Appel, Christos Faloutsos, Jure Leskovec, SDM'10
- Naively: diameter needs **O(N**2)** space and up to O(N**3) time – **prohibitive** (N~1B)
- Our HADI: linear on E (~10B)
  - Near-linear scalability wrt # machines
  - Several optimizations -> 5x faster

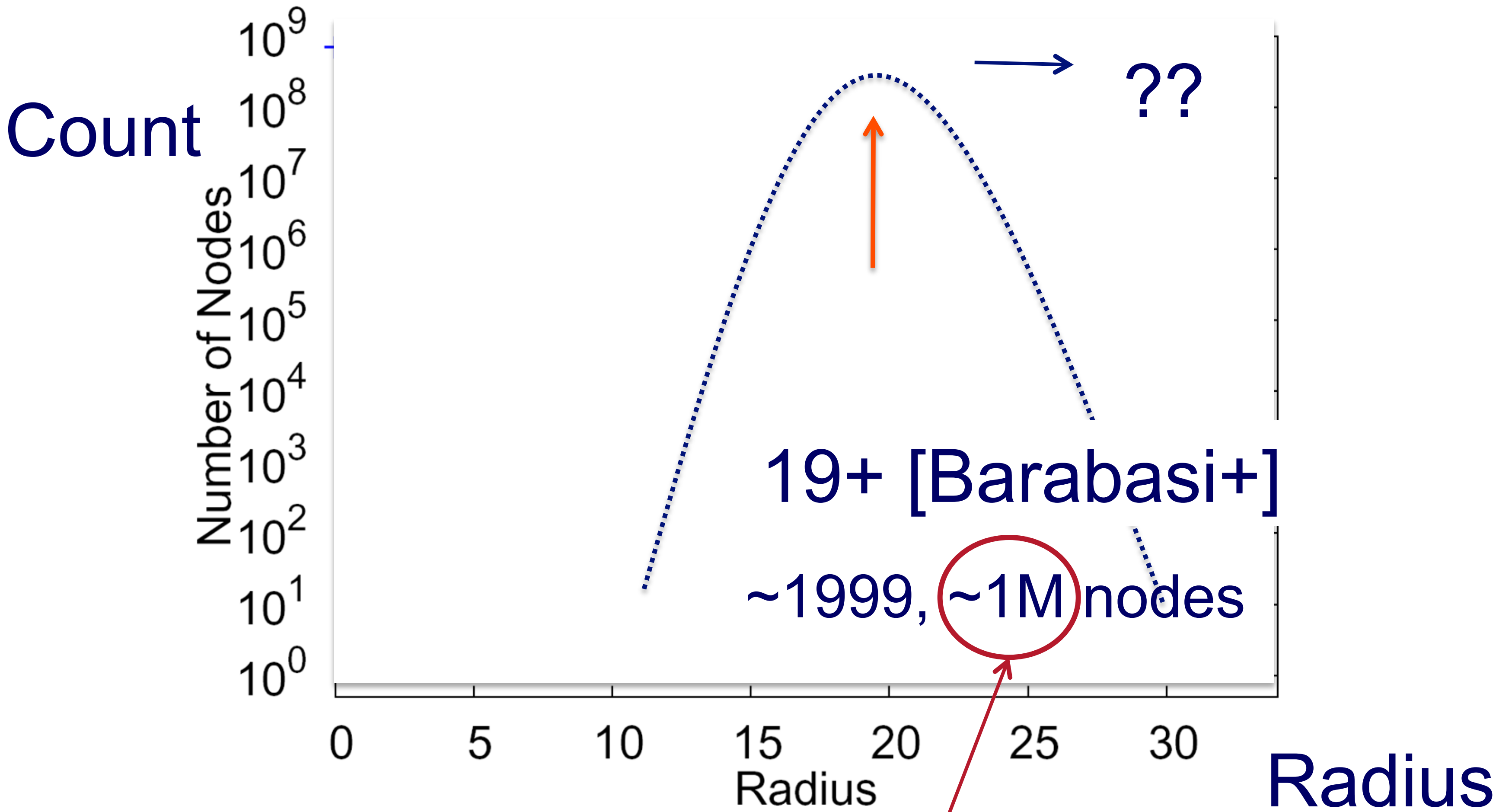
Count



19+ [Barabasi+]

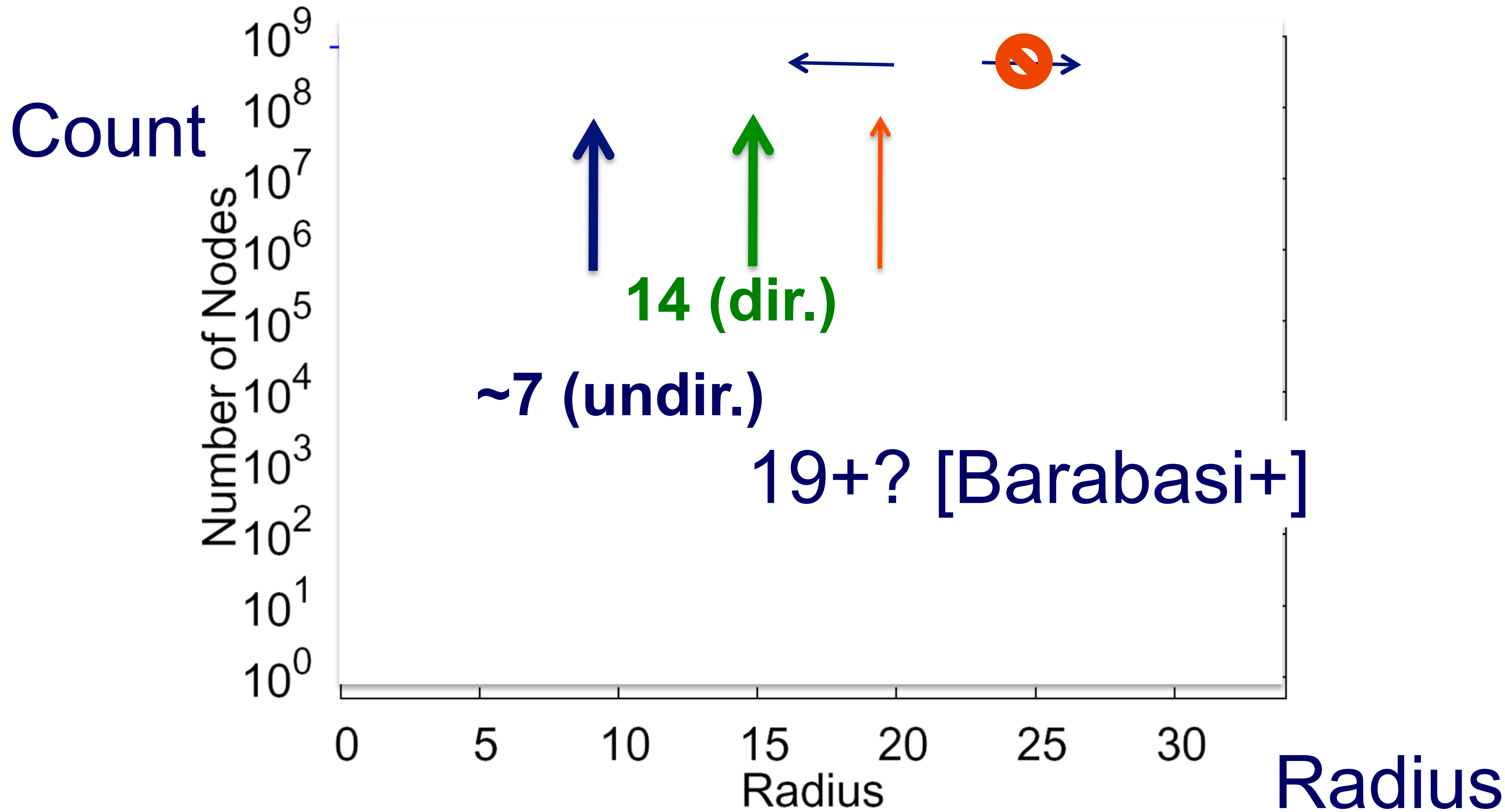
~1999, ~1M nodes

Radius



YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

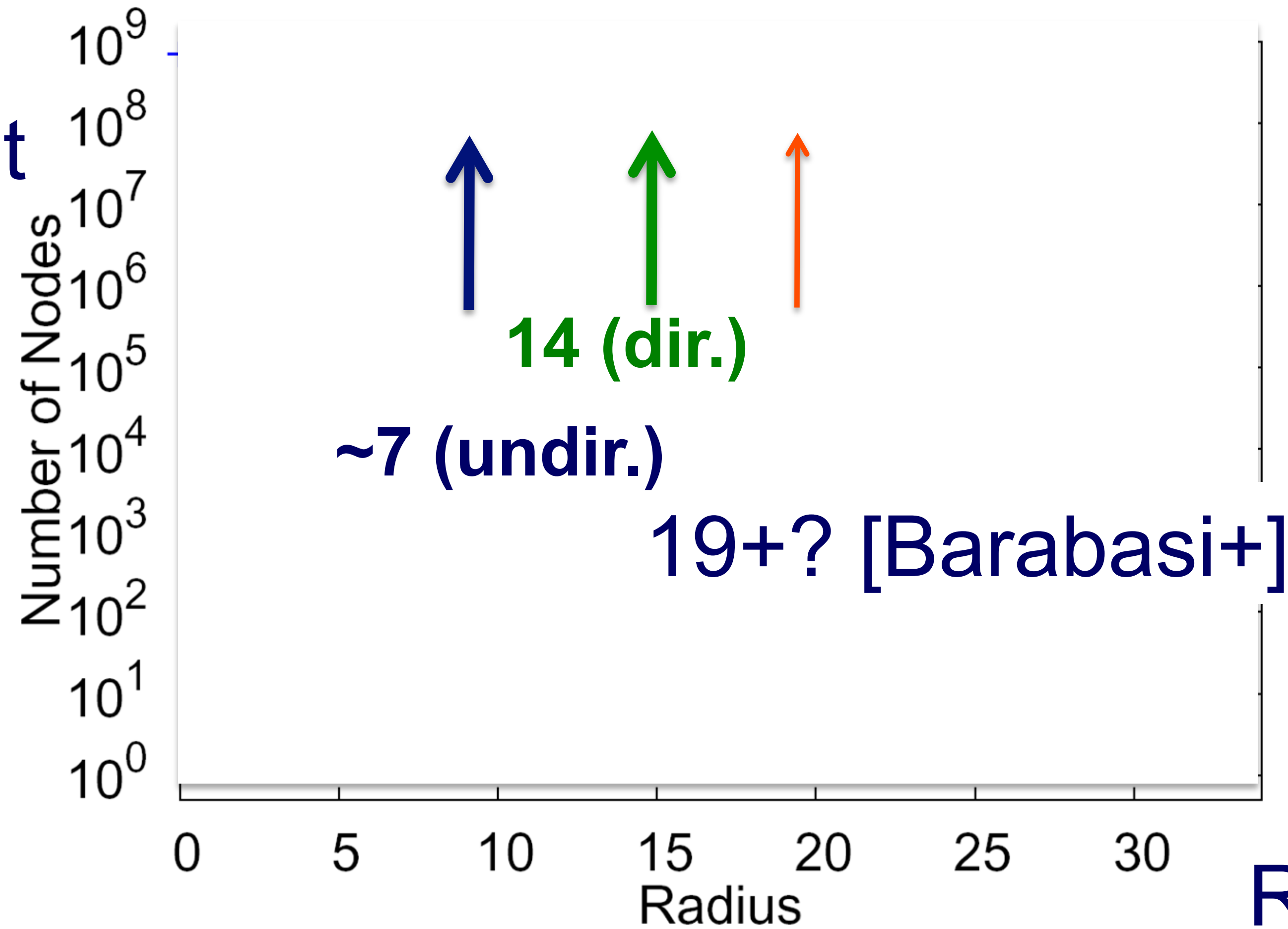
- Largest publicly available graph ever studied.



YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- Largest publicly available graph ever studied.

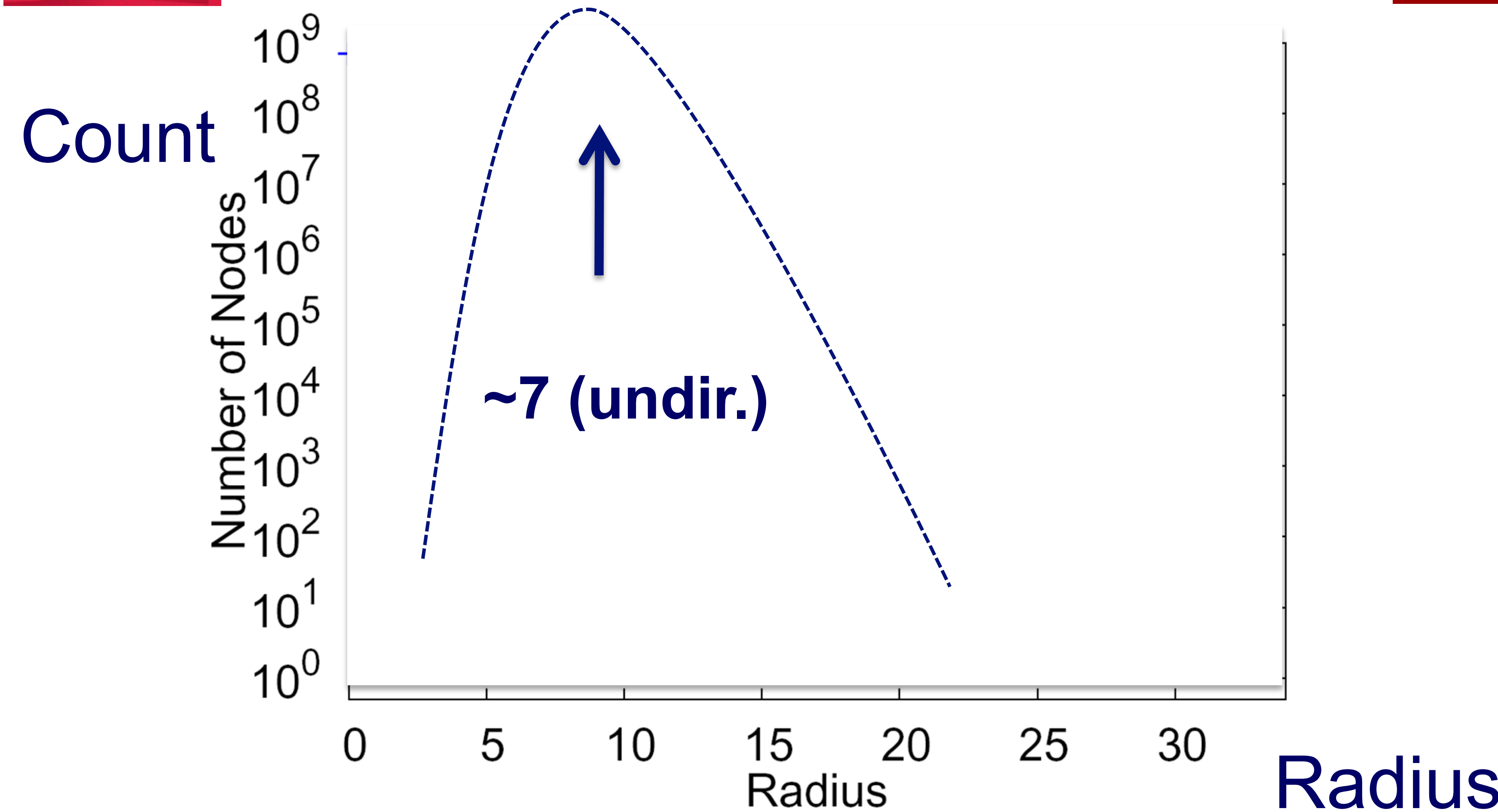
Count



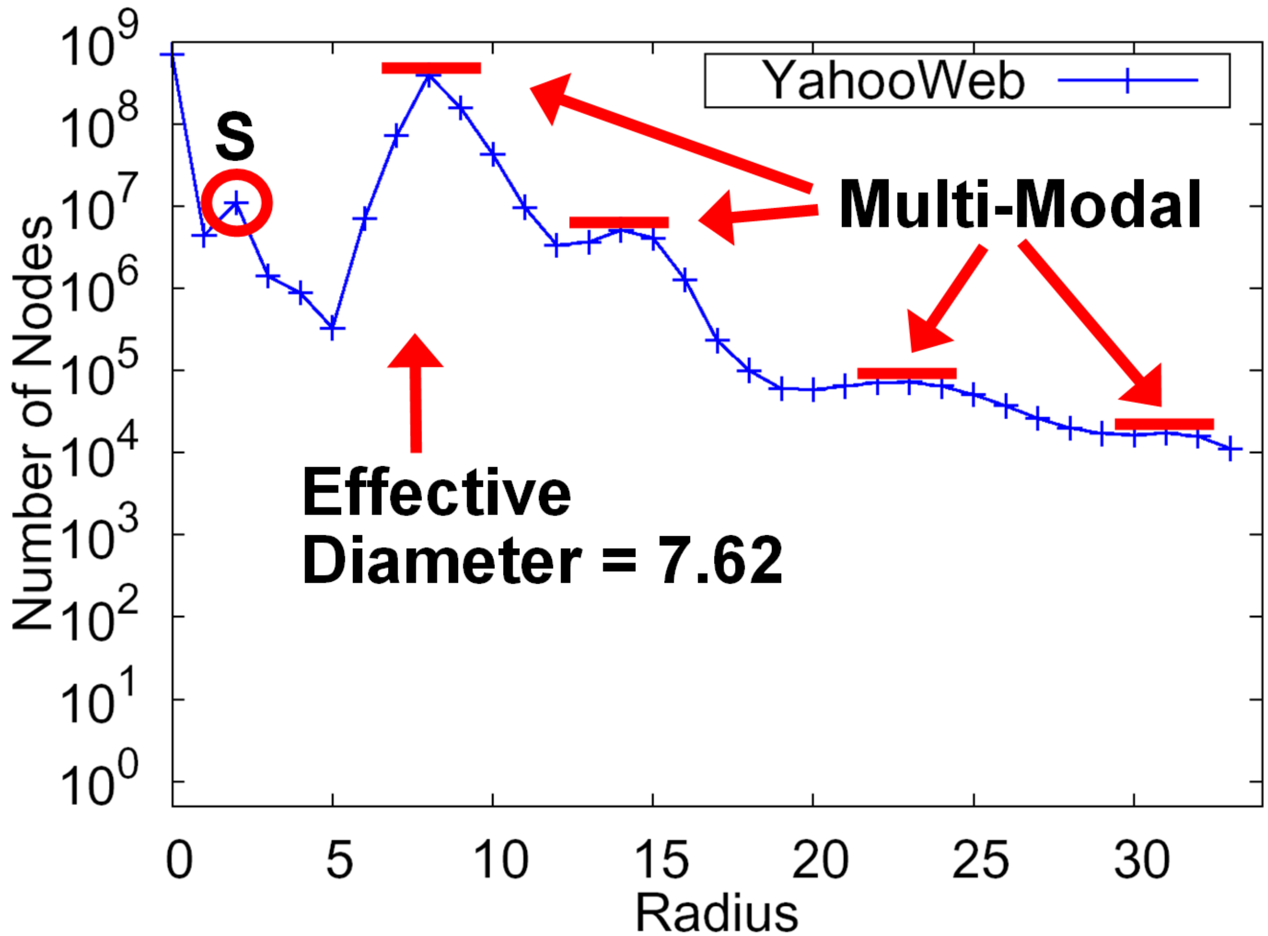
Radius

YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- 7 degrees of separation (!)
- Diameter: shrunk

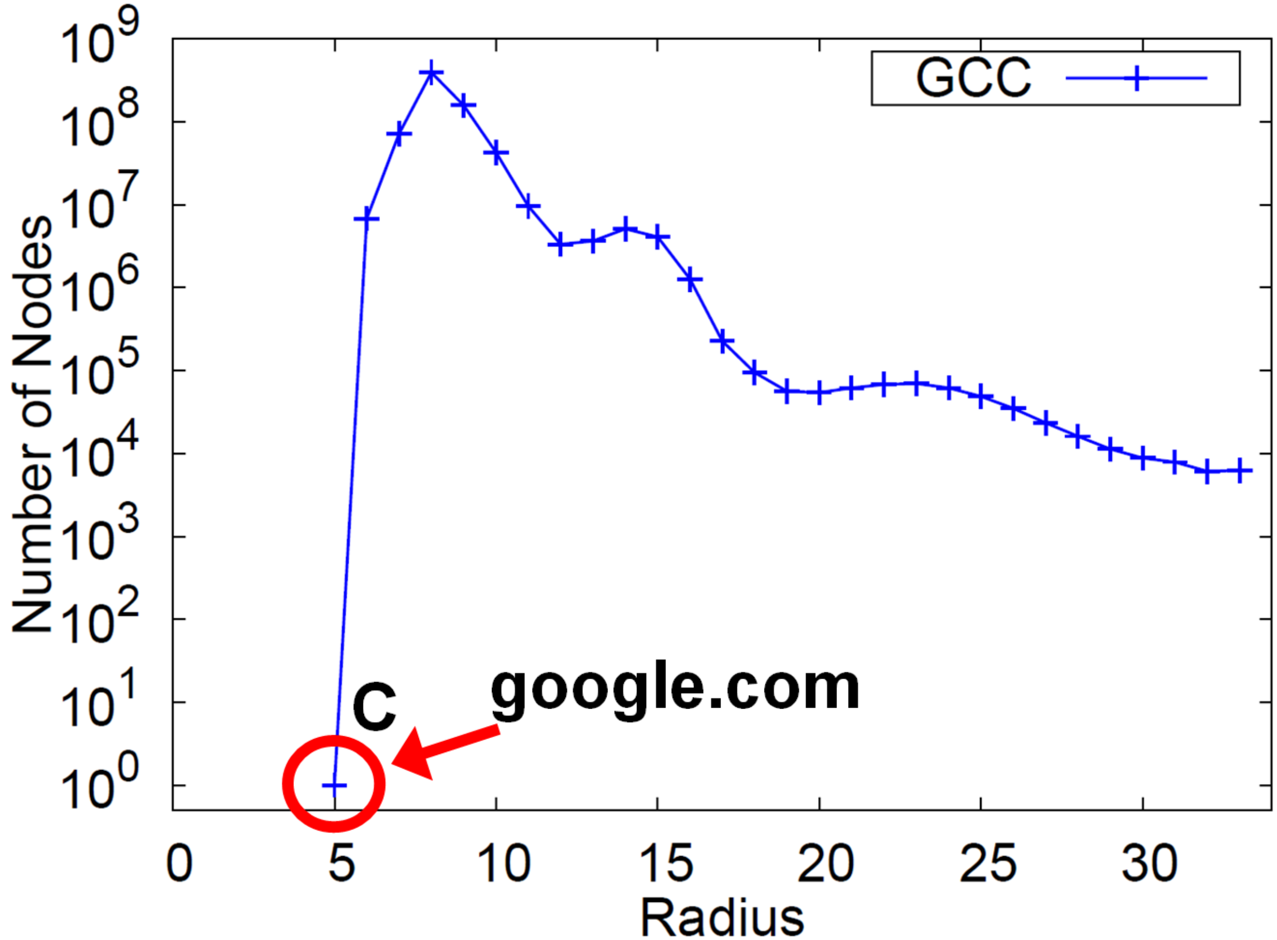


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
Q: Shape?

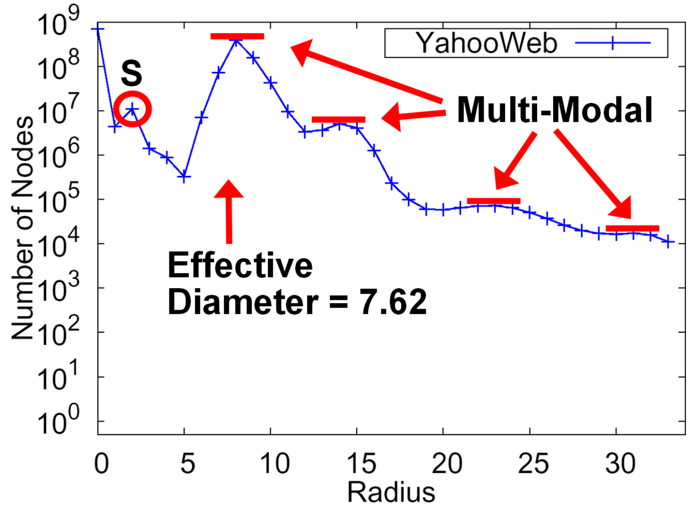


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- effective diameter: surprisingly small.
- Multi-modality (?!)

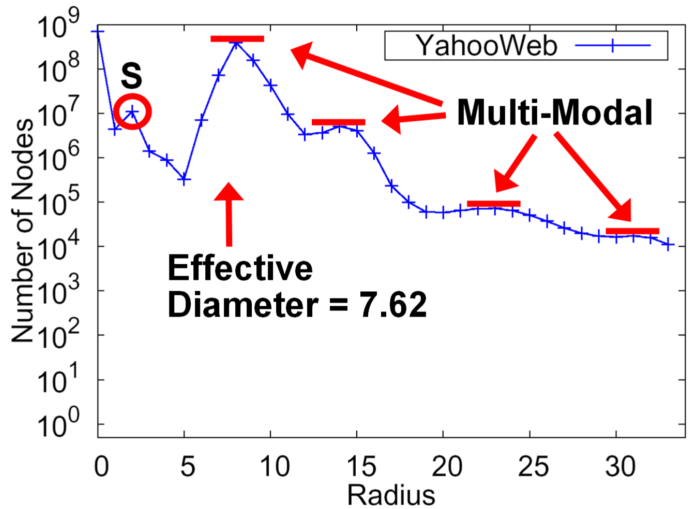


Radius Plot of **GCC** of YahooWeb.

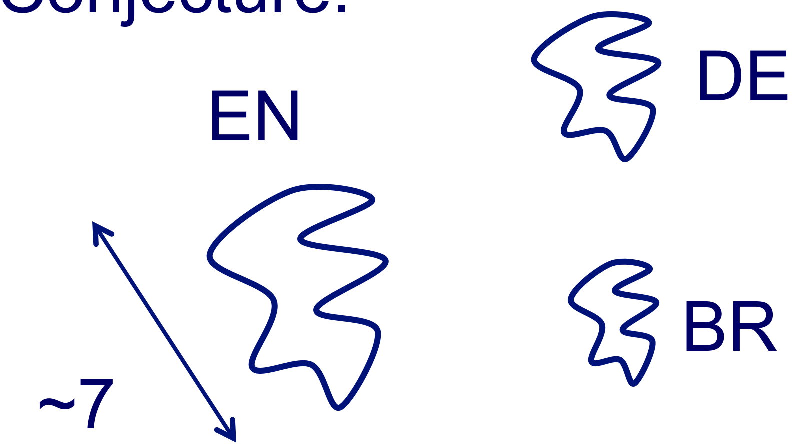


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores .

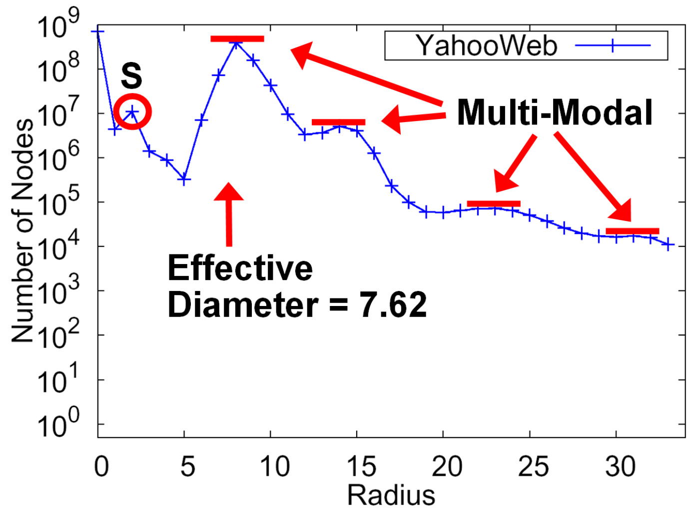


Conjecture:

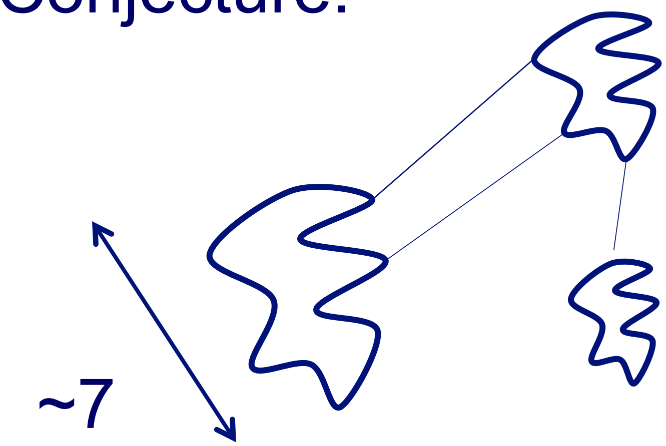


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores .



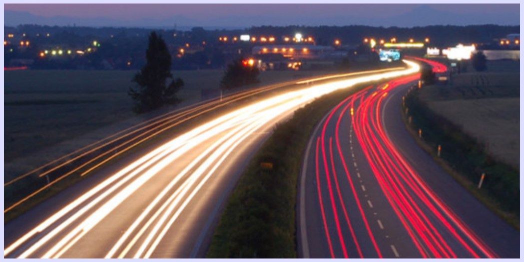
Conjecture:



YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores .

Roadmap



- Patterns in graphs
  - overview
  - Static graphs
    - S1: Degree, S2: eigenvalues
    - S3-4: Triangles, S5: cliques
    - Radius plot
    - Other observations (‘eigenSpokes’)
  - Weighted graphs
  - Time-evolving graphs

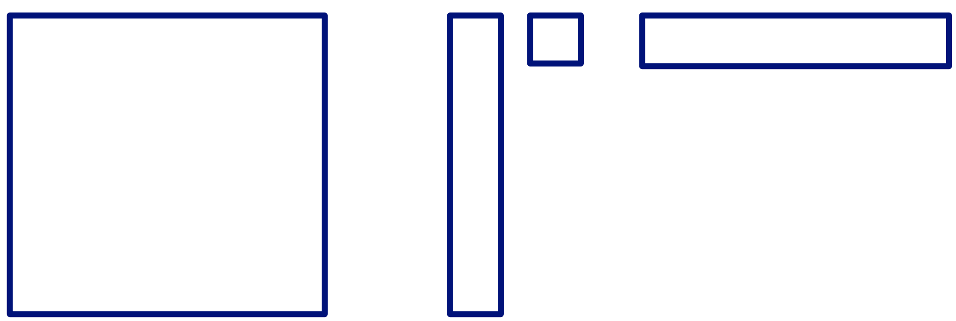
S6: EigenSpokes

B. Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju and Christos Faloutsos: *EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs*, PAKDD 2010, Hyderabad, India, 21-24 June 2010.

EigenSpokes

- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

$$A = U \Sigma U^T$$

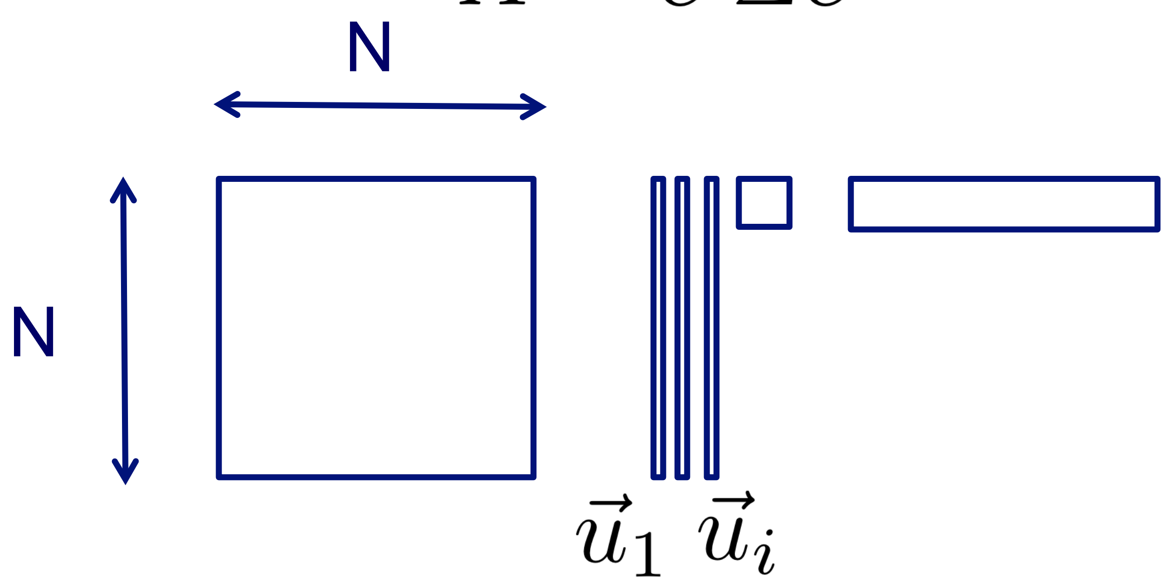


details

EigenSpokes

- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

$$A = U \Sigma U^T$$

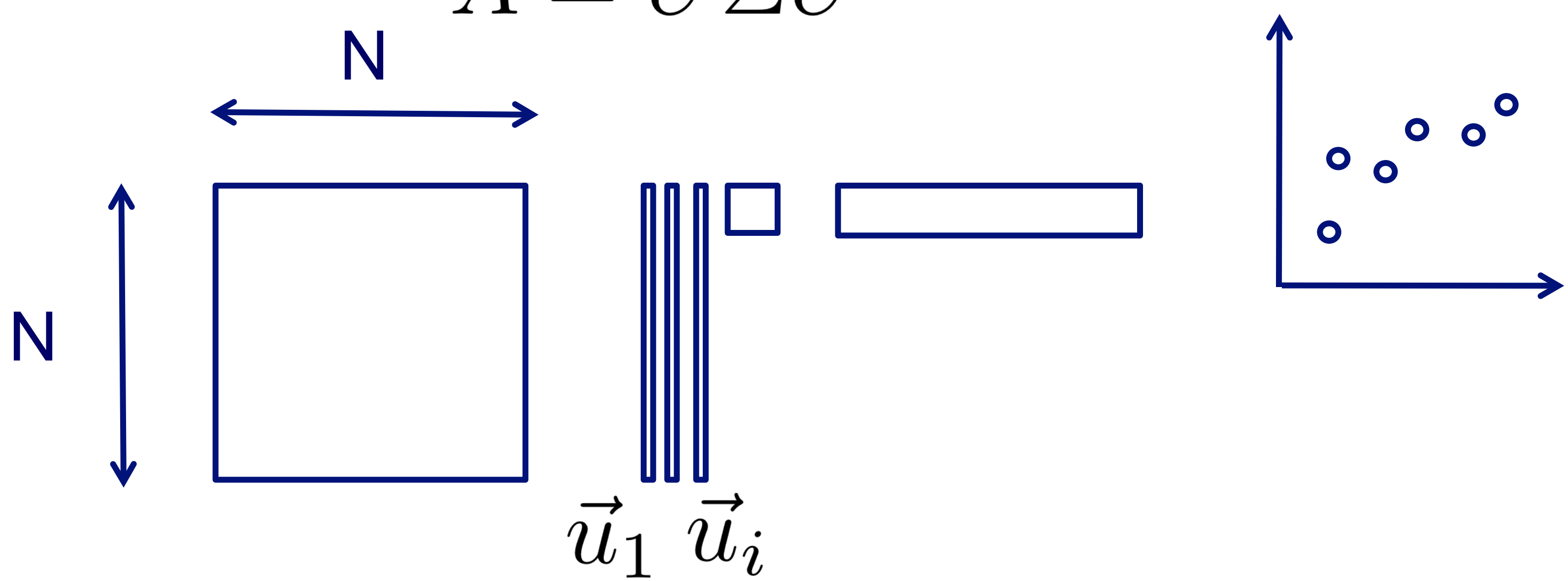


details

EigenSpokes

- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

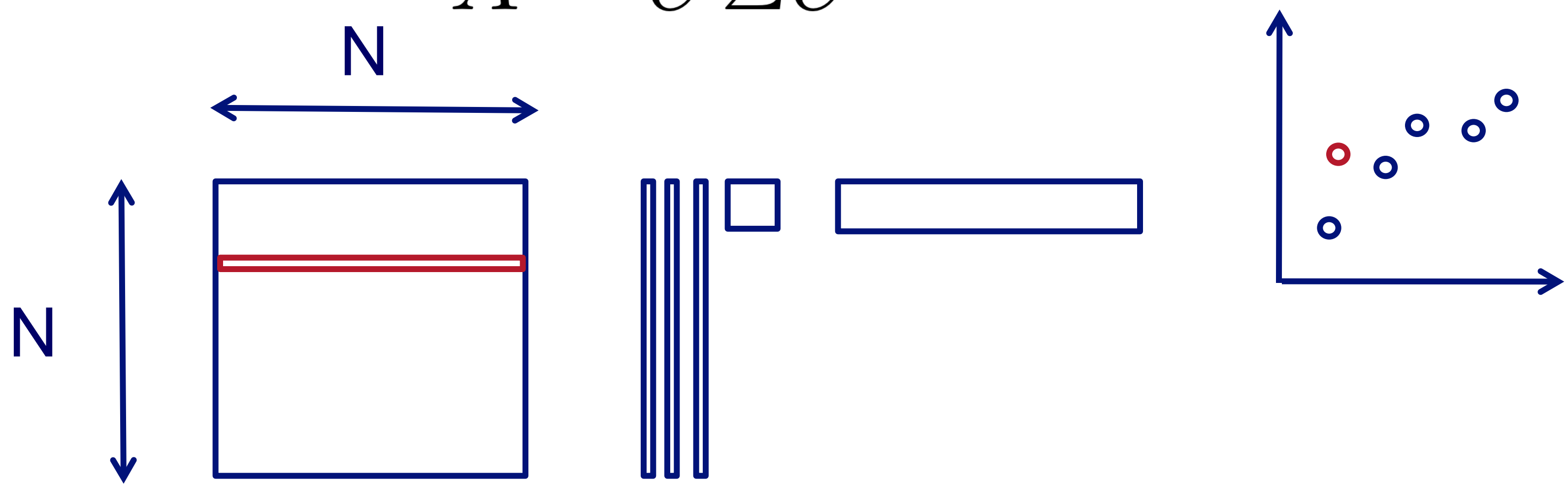
$$A = U \Sigma U^T$$



EigenSpokes

- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

$$A = U \Sigma U^T$$



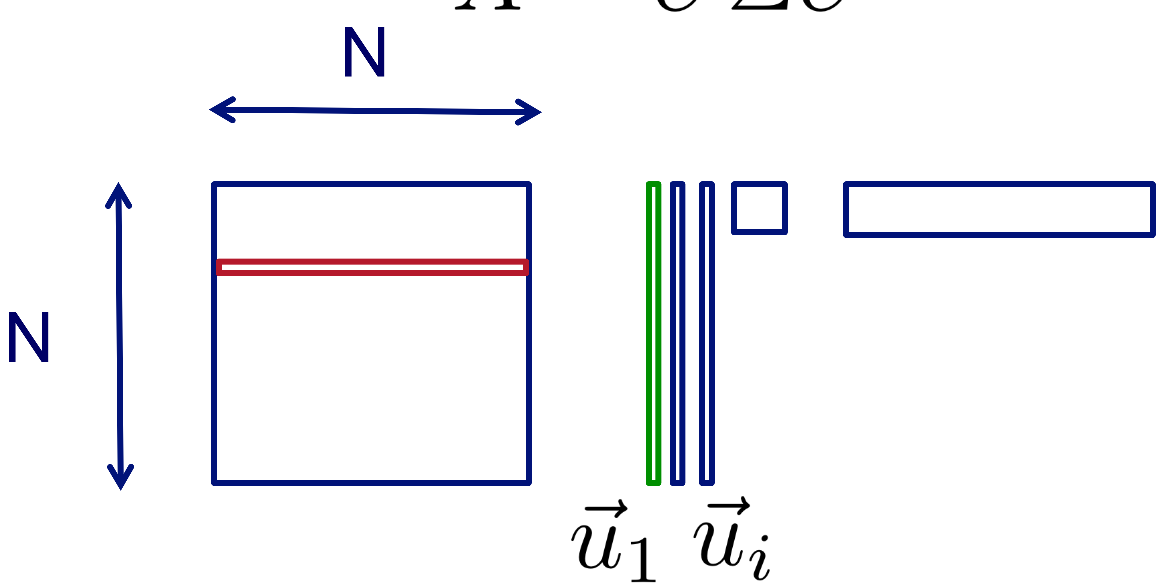
$\vec{u}_1 \quad \vec{u}_i$

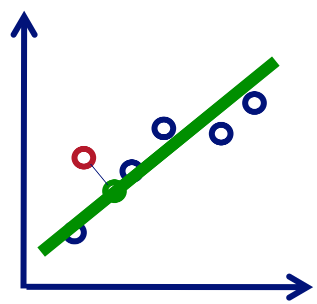
details

EigenSpokes

- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

$$A = U \Sigma U^T$$

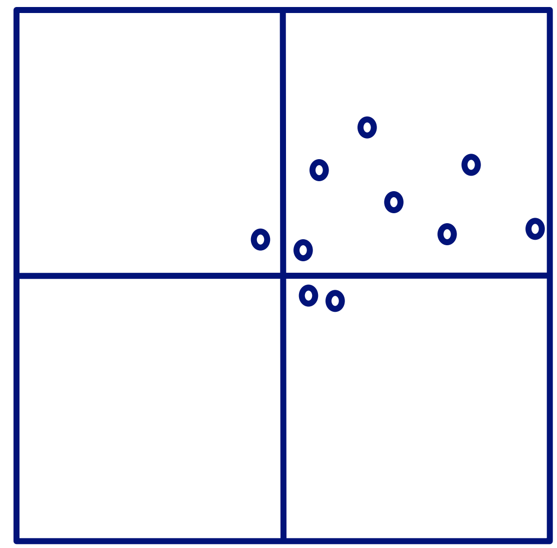




EigenSpokes

- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A few scattered ~randomly

2nd Principal component
u2

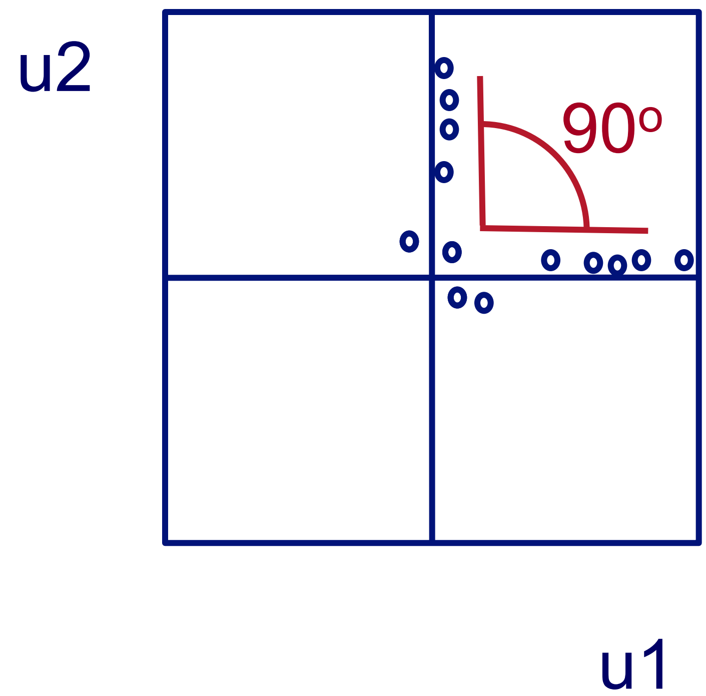


u1

1st Principal component

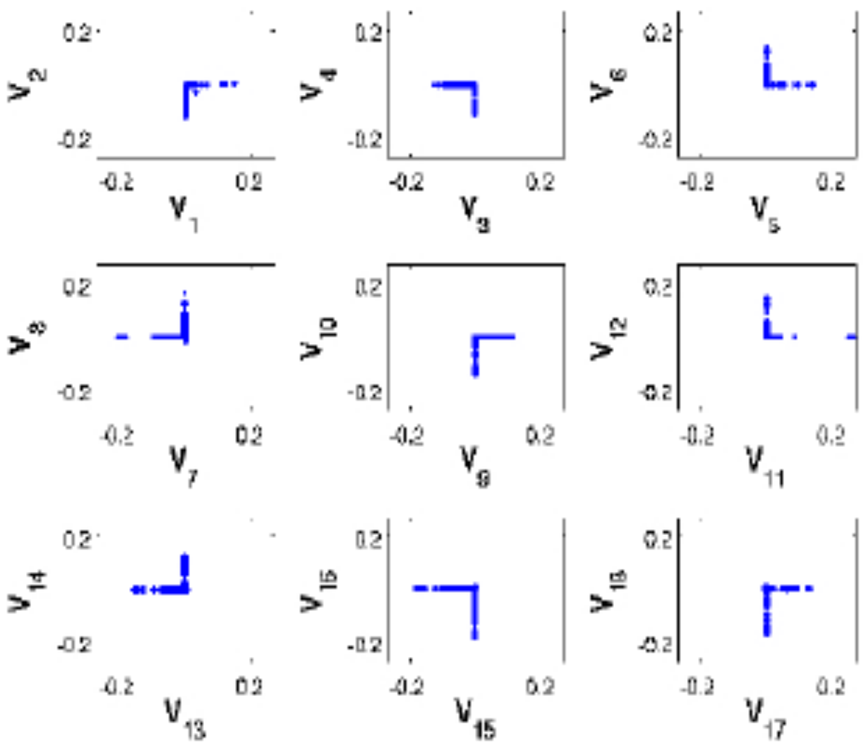
EigenSpokes

- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A f[illegible]ttered ~r[illegible]d[illegible]mly



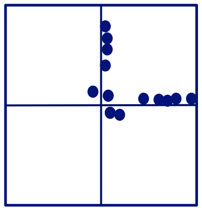
EigenSpokes - pervasiveness

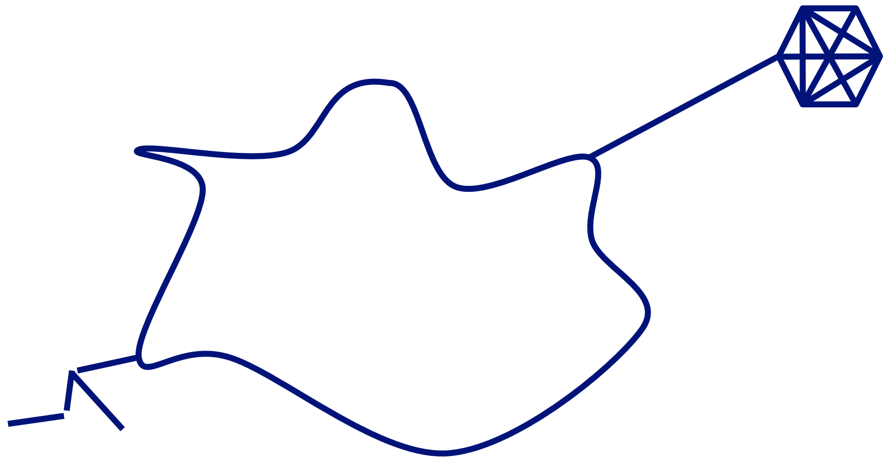
- Present in mobile social graph
  - across time and space
- Patent citation graph



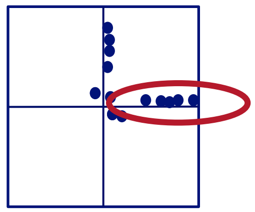
EigenSpokes - explanation

Near-cliques, or near-bipartite-cores, loosely connected

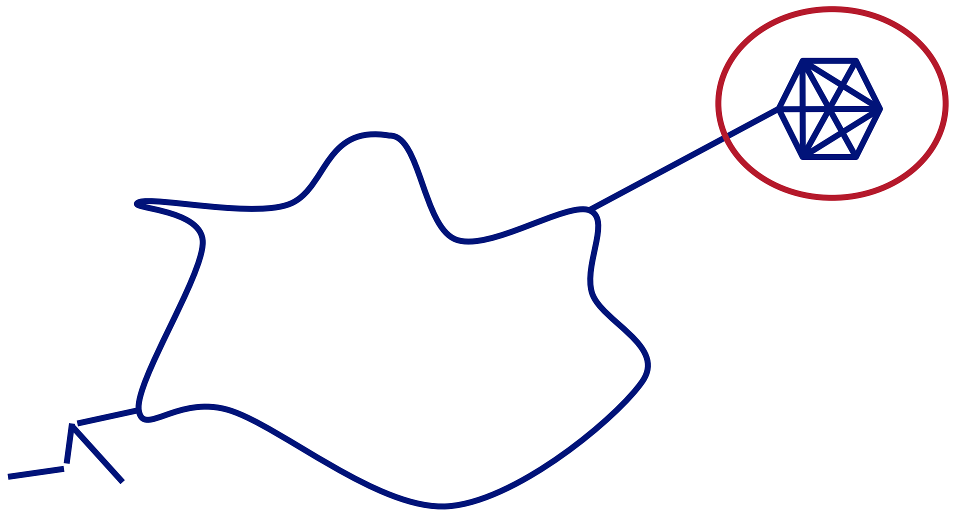




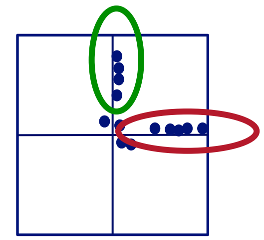
EigenSpokes - explanation



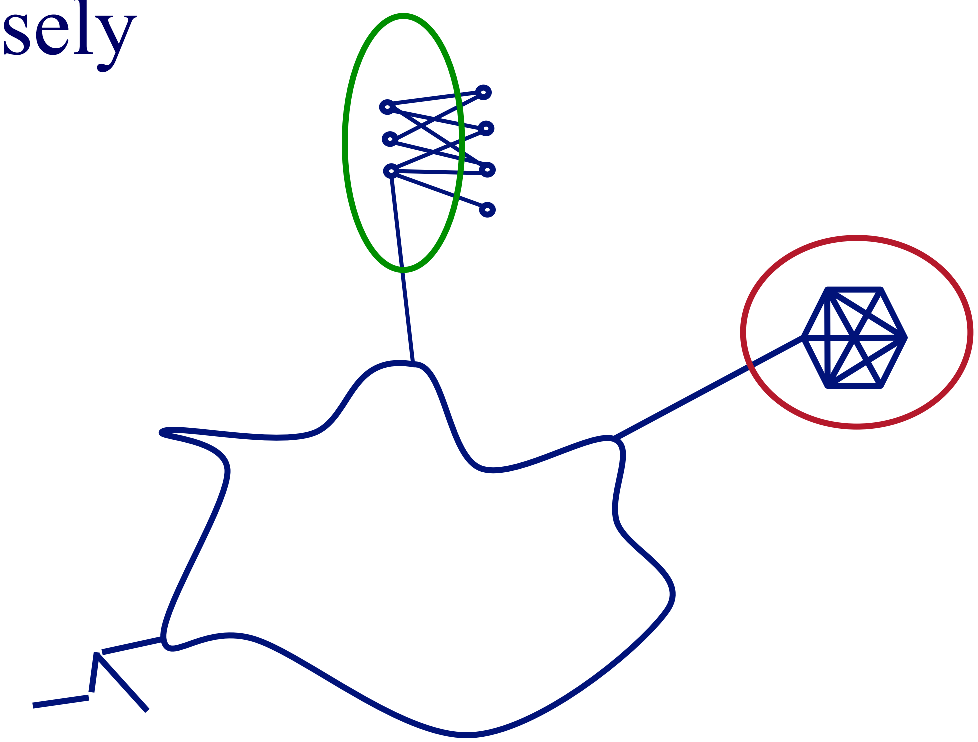
Near-cliques, or near-bipartite-cores, loosely connected



EigenSpokes - explanation

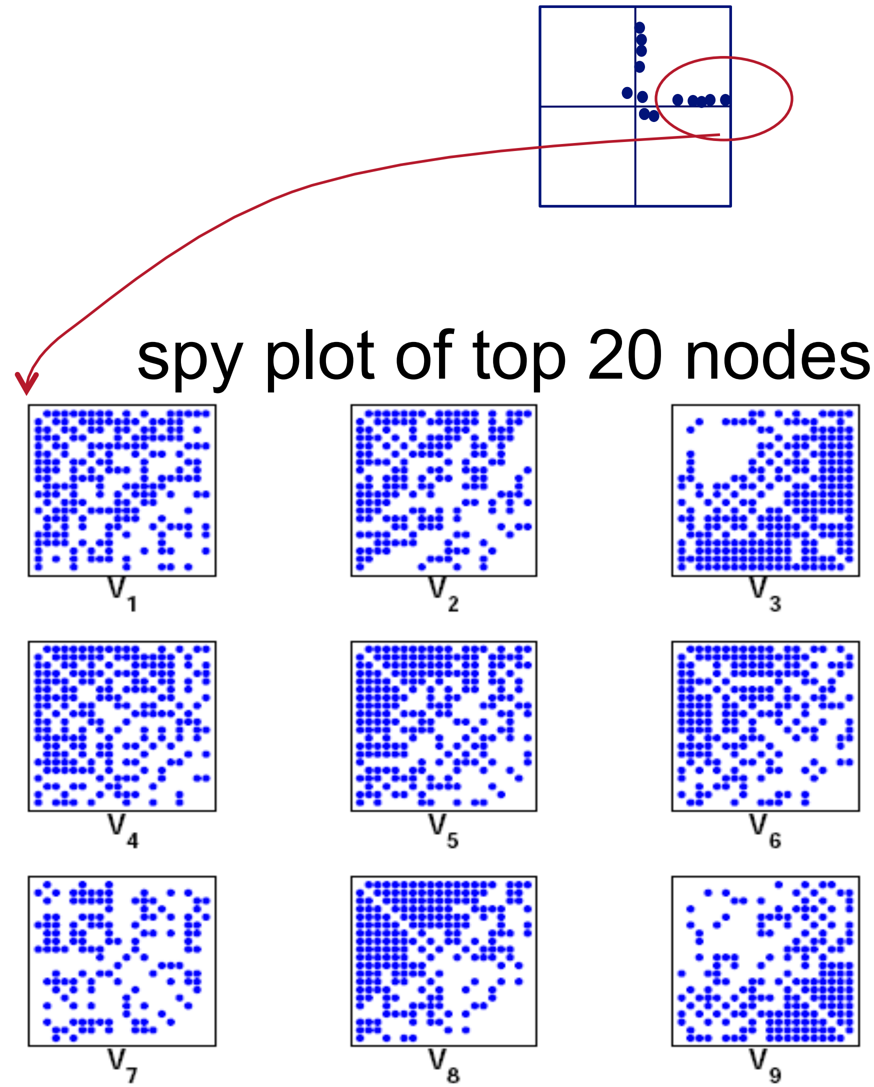


Near-cliques, or near-bipartite-cores, loosely connected



EigenSpokes - explanation

Near-cliques, or near-bipartite-cores, loosely connected



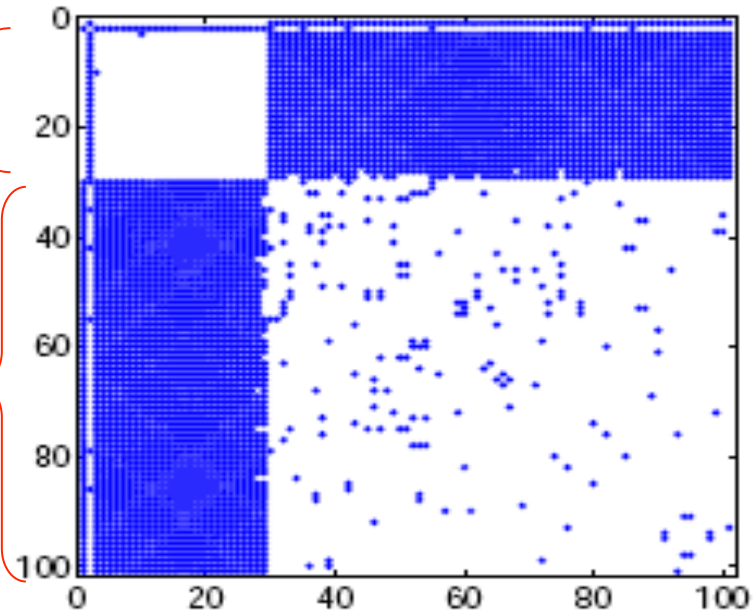
So what?

- Extract nodes with high *scores*
- high connectivity
- Good "communities"

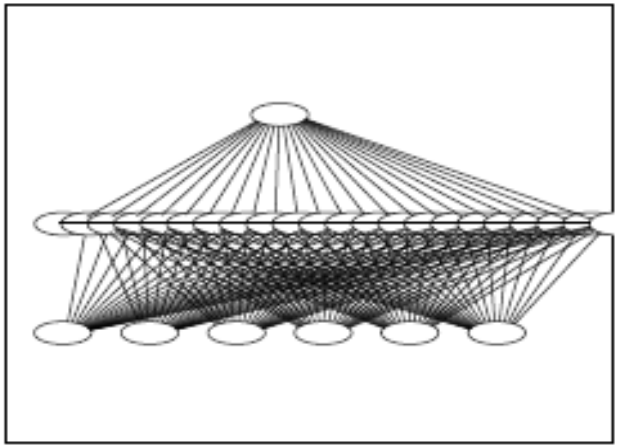
Bipartite Communities!

patents from same inventor(s)

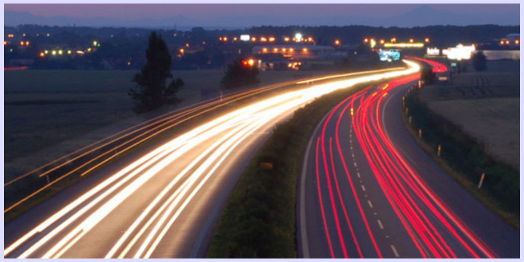
`cut-and-paste' bibliography!



magnified bipartite community



Roadmap



- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

Observations on weighted graphs?

- A: yes - even more 'laws'!

M. McGlohon, L. Akoglu, and C. Faloutsos
Weighted Graphs and Disconnected Components: Patterns and a Generator.
SIG-KDD 2008

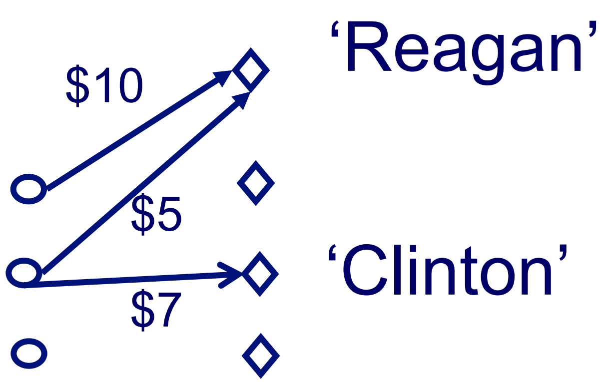
Observation W.1: Fortification

Q: How do the weights

of nodes relate to degree?

Observation W.1: Fortification

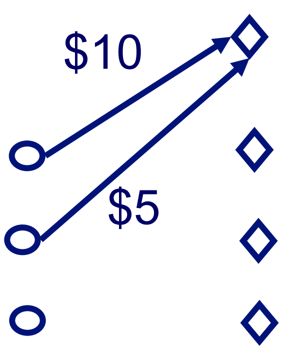
More donors, more $?



Observation W.1: fortification: Snapshot Power Law

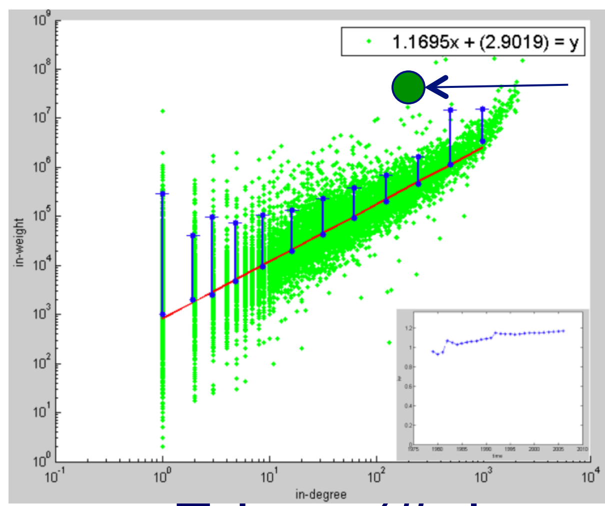
- Weight: super-linear on in-degree
- exponent 'iw': 1.01 < iw < 1.26

More donors,
<u>even</u> more $



Orgs-Candidates

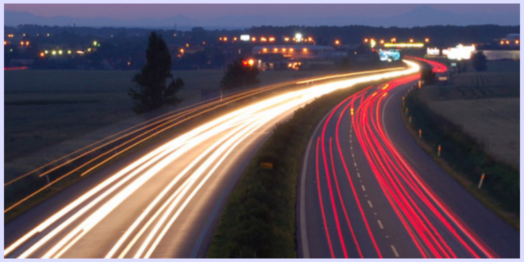
In-weights ($)



Edges (# donors)

e.g. John Kerry, $10M received, from 1K donors

Roadmap

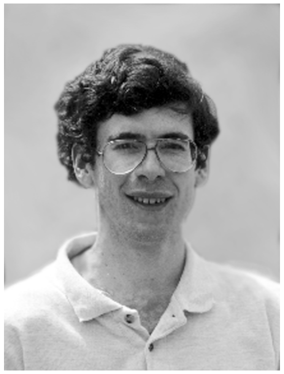


- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

Problem: Time evolution

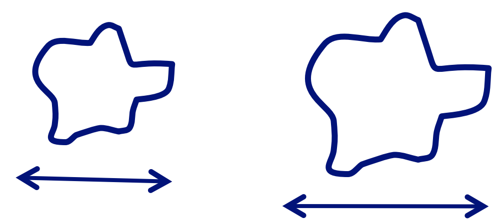
- with Jure Leskovec (CMU -> Stanford)

- and Jon Kleinberg (Cornell – sabb. @ CMU)



T.1 Evolution of the Diameter

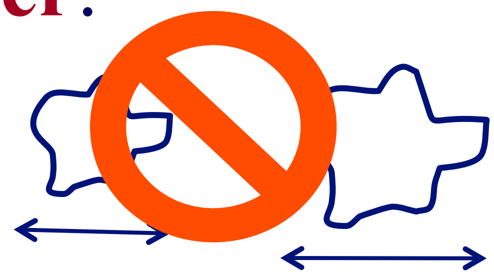
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)



- What is happening in real data?

T.1 Evolution of the Diameter

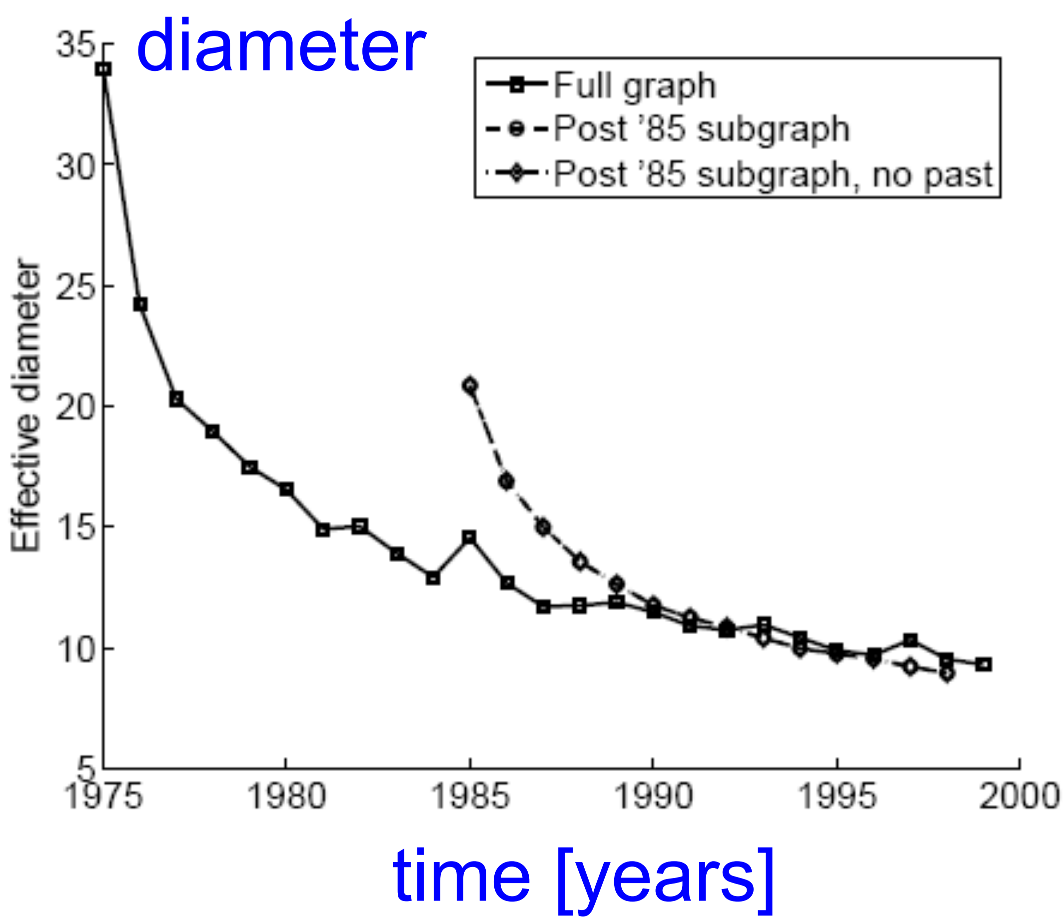
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)



- What is happening in real data?
- Diameter **shrinks** over time

T.1 Diameter – “Patents”

- Patent citation network
- 25 years of data
- @1999
  - 2.9 M nodes
  - 16.5 M edges



T.2 Temporal Evolution of the Graphs

- $N(t)$ … nodes at time t
- $E(t)$ … edges at time t
- Suppose that

  $$N(t+1) = 2 * N(t)$$

- Q: what is your guess for

  $$E(t+1) =? \; 2 * E(t)$$

T.2 Temporal Evolution of the Graphs

- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

$$N(t+1) = 2 * N(t)$$

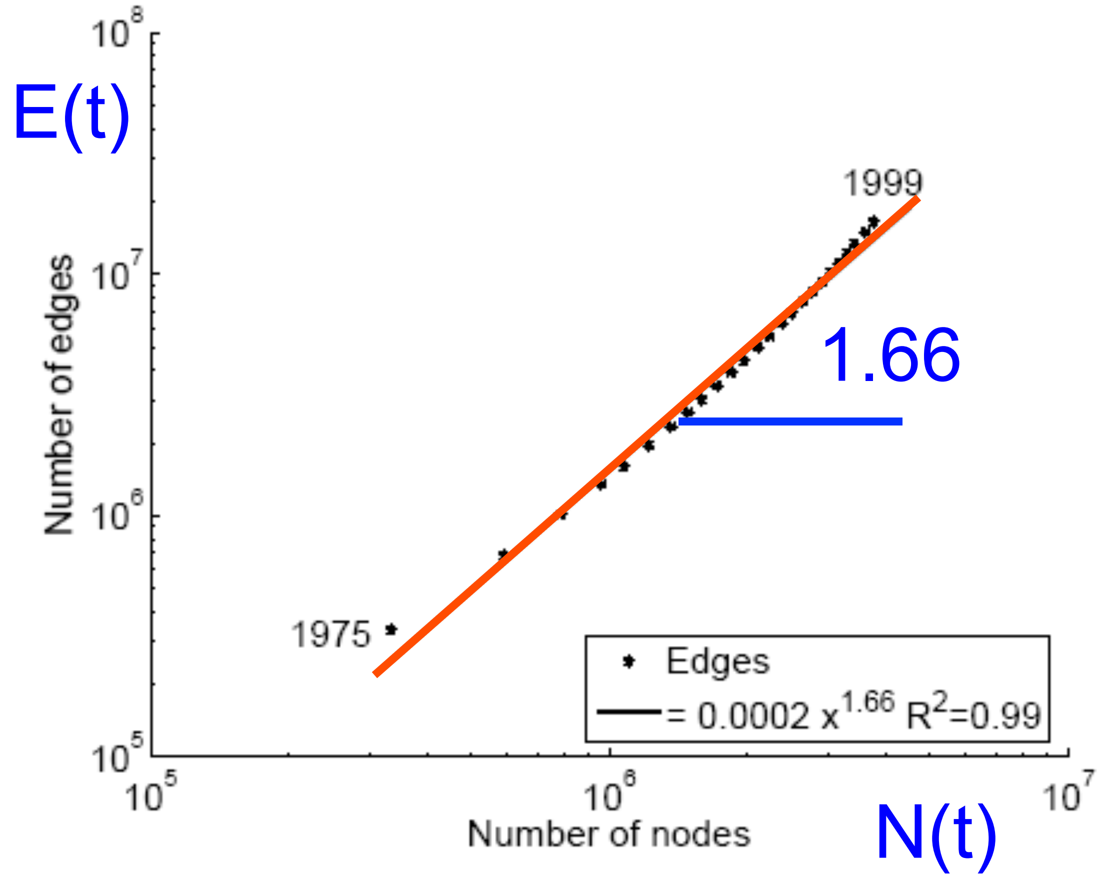
- Q: what is your guess for

$$E(t+1) \cancel{=?\ 2} * E(t)$$

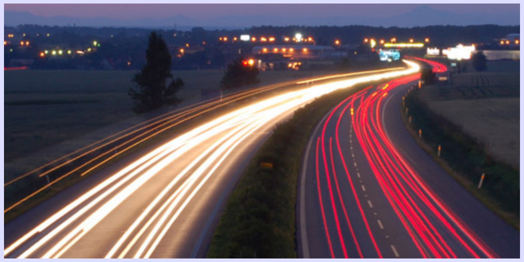
- A: over-doubled!
  – But obeying the ``Densification Power Law''

T.2 Densification – Patent Citations

- Citations among patents granted
- @1999
  - 2.9 M nodes
  - 16.5 M edges
- Each year is a datapoint



Roadmap



- Patterns in graphs
  - …
  - Time-evolving graphs
    - T1: shrinking diameter;
    - T2: densification
    - T3: connected components
    - T4: popularity over time
    - T5: phonecall patterns
- …

More on Time-evolving graphs

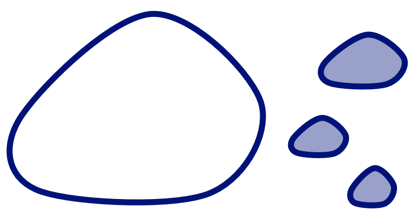
M. McGlohon, L. Akoglu, and C. Faloutsos
Weighted Graphs and Disconnected Components: Patterns and a Generator.
SIG-KDD 2008

Observation T.3: NLCC behavior

Q: How do NLCC's emerge and join with the GCC?

(``NLCC'' = non-largest conn. components)

- Do they continue to grow in size?
- or do they shrink?
- or stabilize?



Observation T.3: NLCC behavior

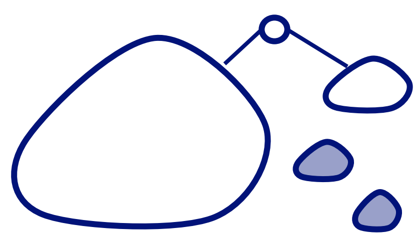
Q: How do NLCC's emerge and join with the GCC?

(``NLCC'' = non-largest conn. components)

– Do they continue to grow in size?

– or do they <u>shrink</u>?

– or stabilize?



Observation T.3: NLCC behavior

Q: How do NLCC's emerge and join with the GCC?

(``NLCC'' = non-largest conn. components)

YES – Do they continue to grow in size?

YES – or do they shrink?

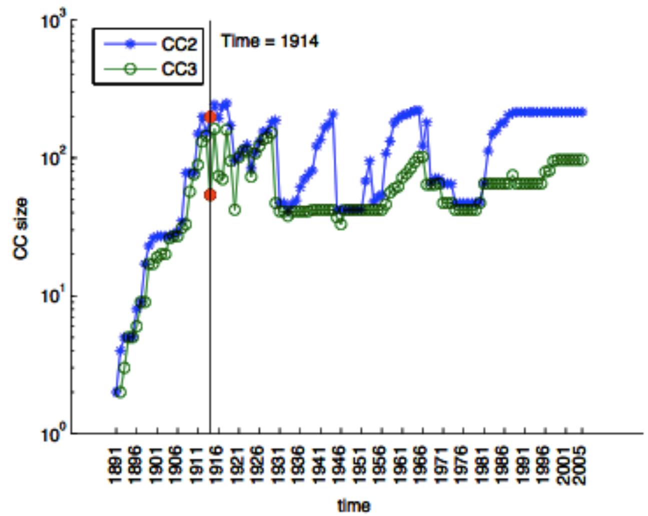
YES – or stabilize?

Observation T.3: NLCC behavior

- After the gelling point, the GCC takes off, but NLCC's remain ~constant (actually, **oscillate**).

IMDB

CC size



Time-stamp

(Computation – scalability?)

- Q: How to handle billion node graphs?
- A: hadoop + ‘Pegasus’
  - Most operations -> matrix-vector multiplications

Generalized Iterated Matrix Vector Multiplication (GIMV)

PEGASUS: A Peta-Scale Graph Mining System - Implementation and Observations. U Kang, Charalampos E. Tsourakakis, and Christos Faloutsos. (ICDM) 2009, Miami, Florida, USA. Best Application Paper (runner-up).

Generalized Iterated Matrix Vector Multiplication (GIMV)

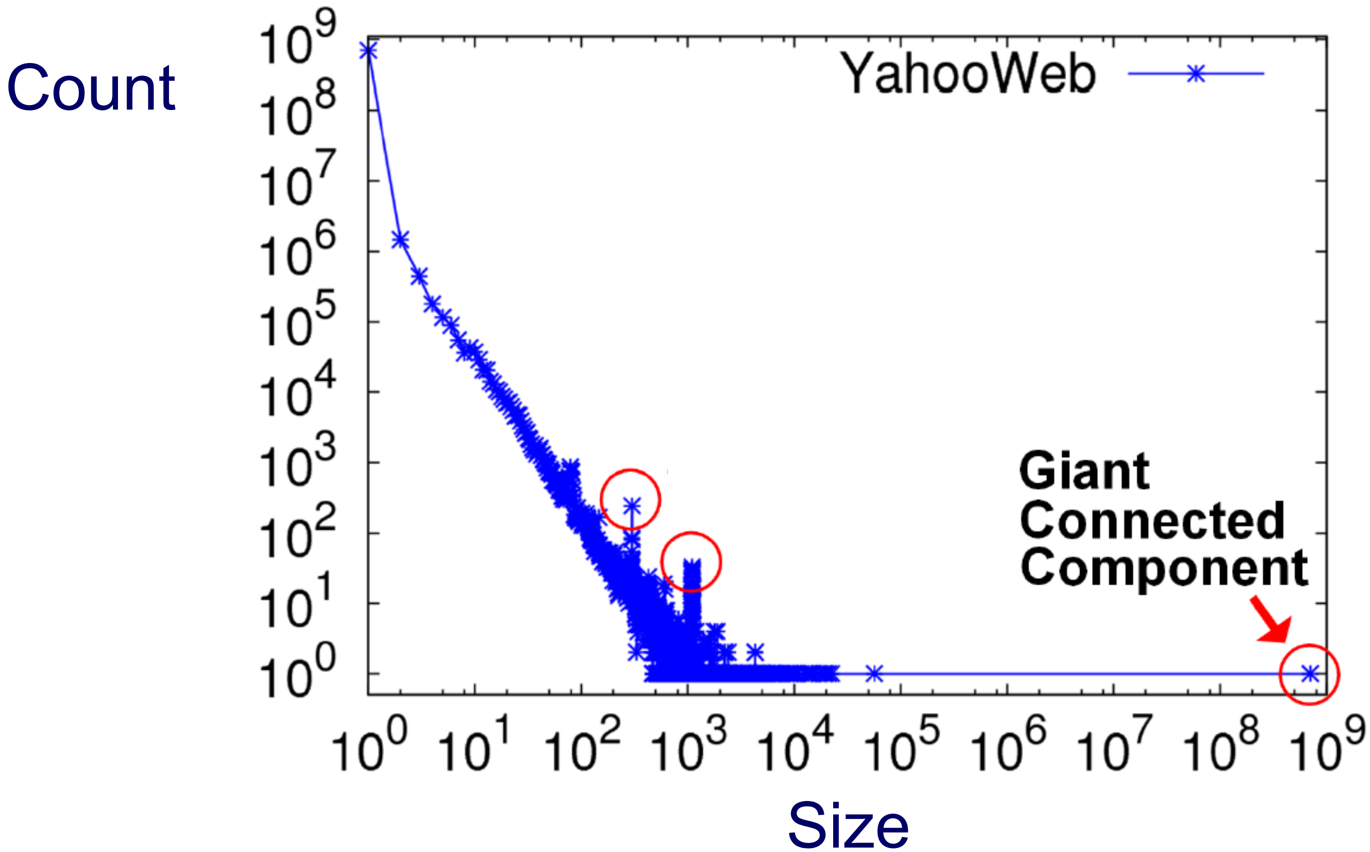
details

- PageRank
- proximity (RWR)
- Diameter
- Connected components
- (eigenvectors,
- Belief Prop.
- …)

Matrix – vector Multiplication (iterated)

Example: GIM-V At Work

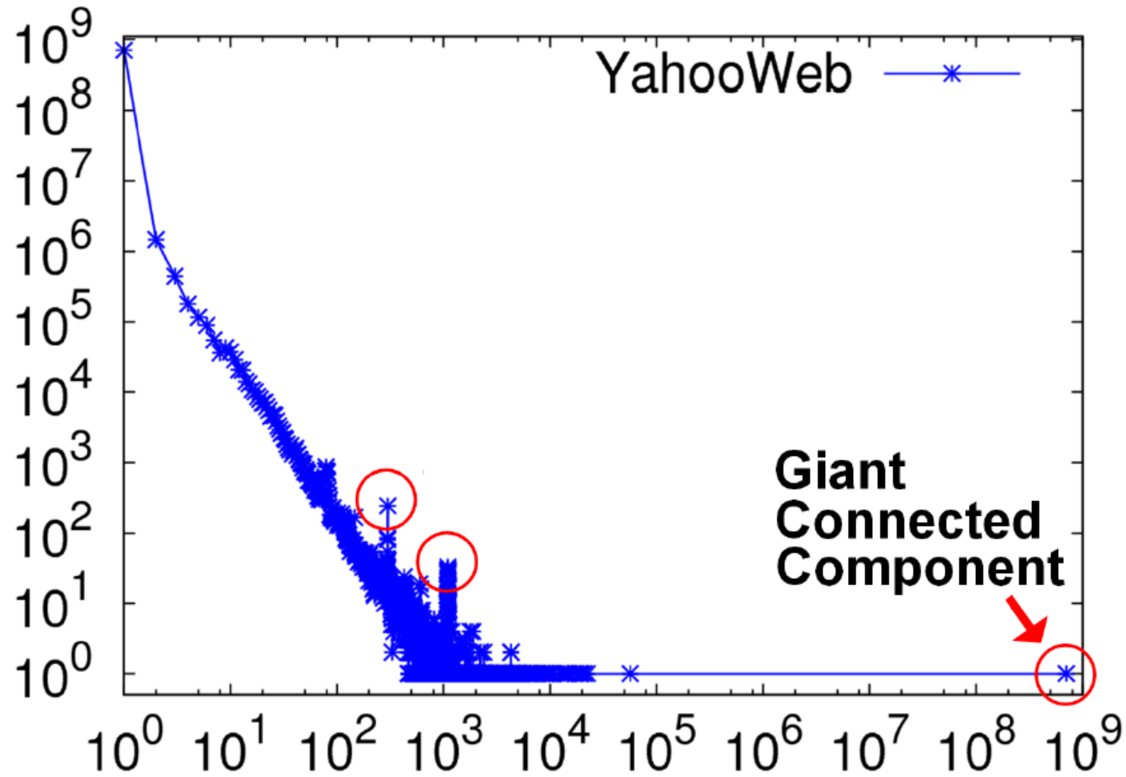
- Connected Components – 4 observations:



Example: GIM-V At Work

- Connected Components

Count



Size

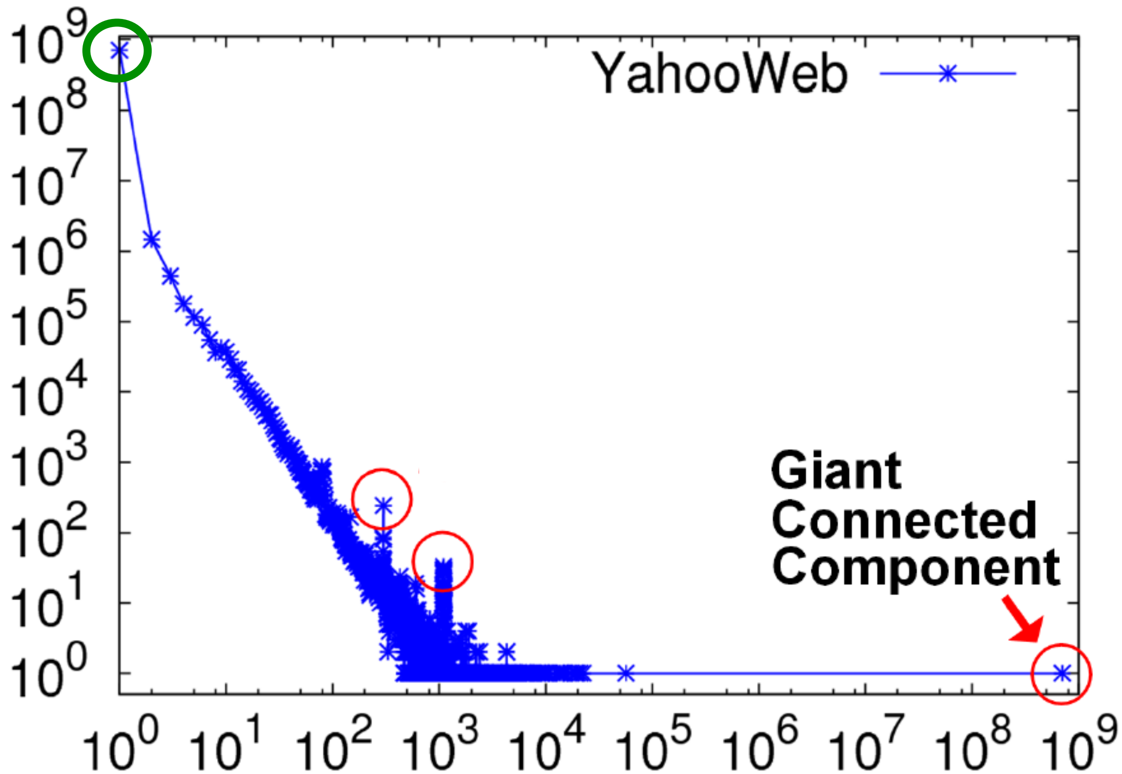
1) 10K x larger than next

Example: GIM-V At Work

- Connected Components

Count

2) ~0.7B singleton nodes



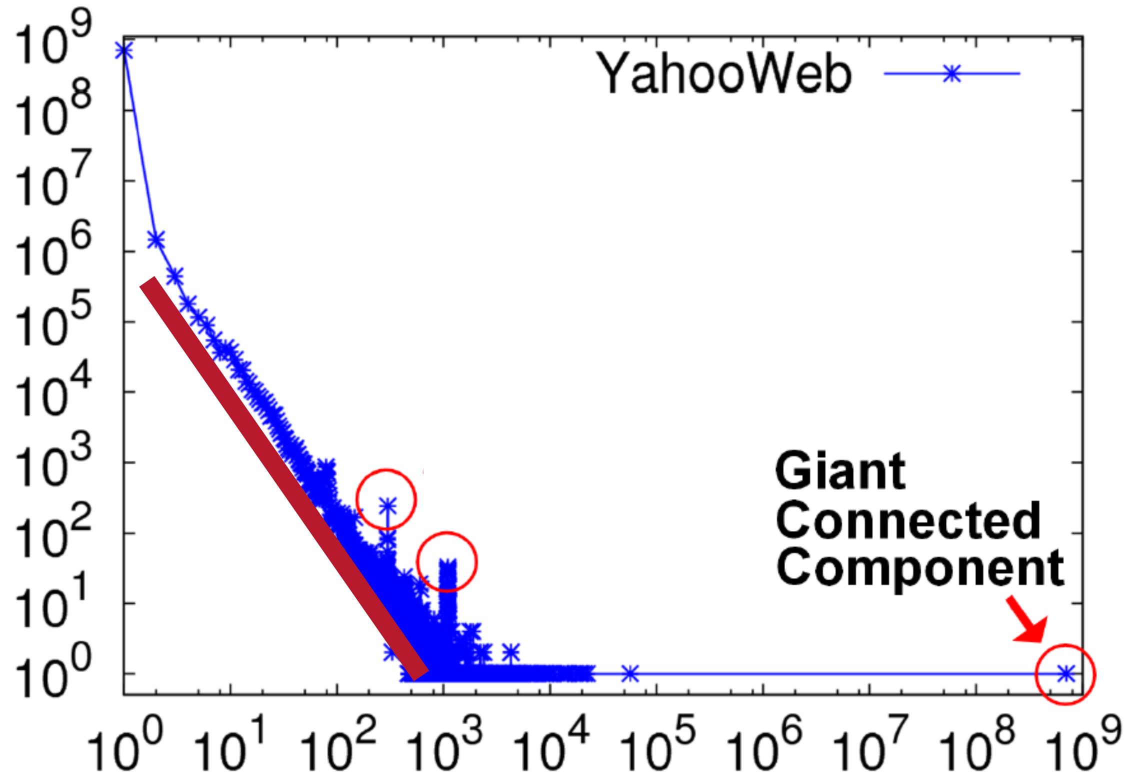
Size

Example: GIM-V At Work

- Connected Components

Count

3) SLOPE!

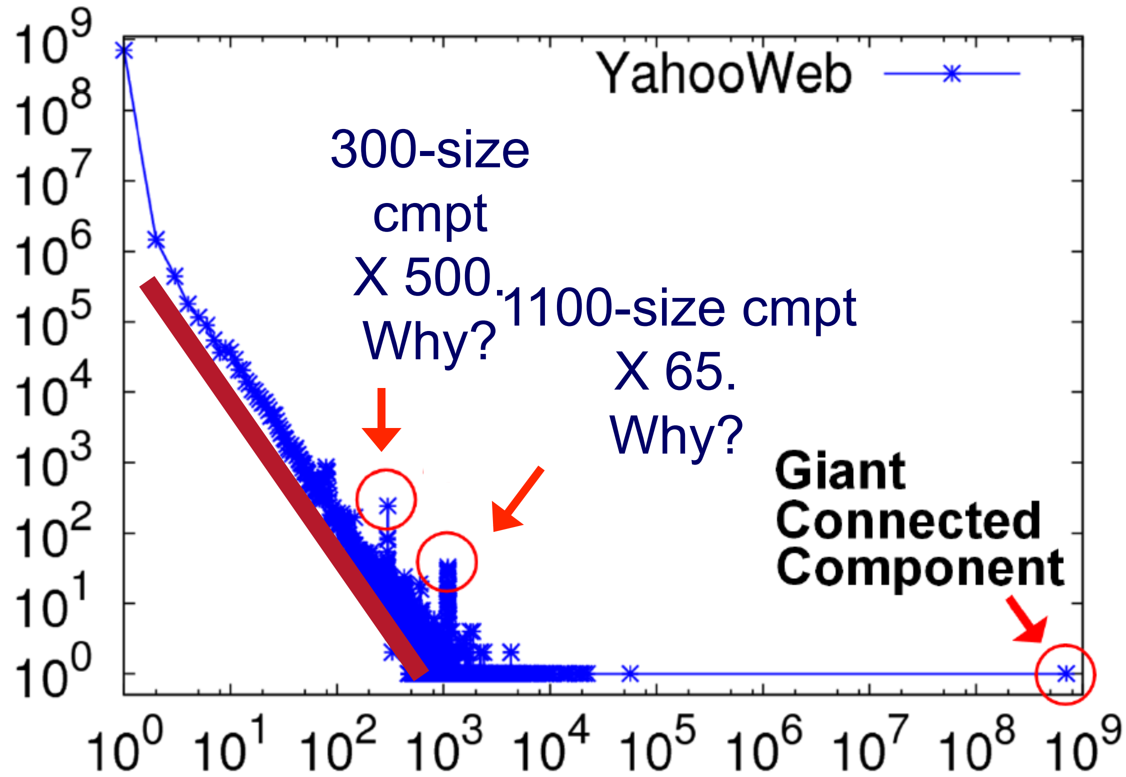


Size

Example: GIM-V At Work

- Connected Components

Count

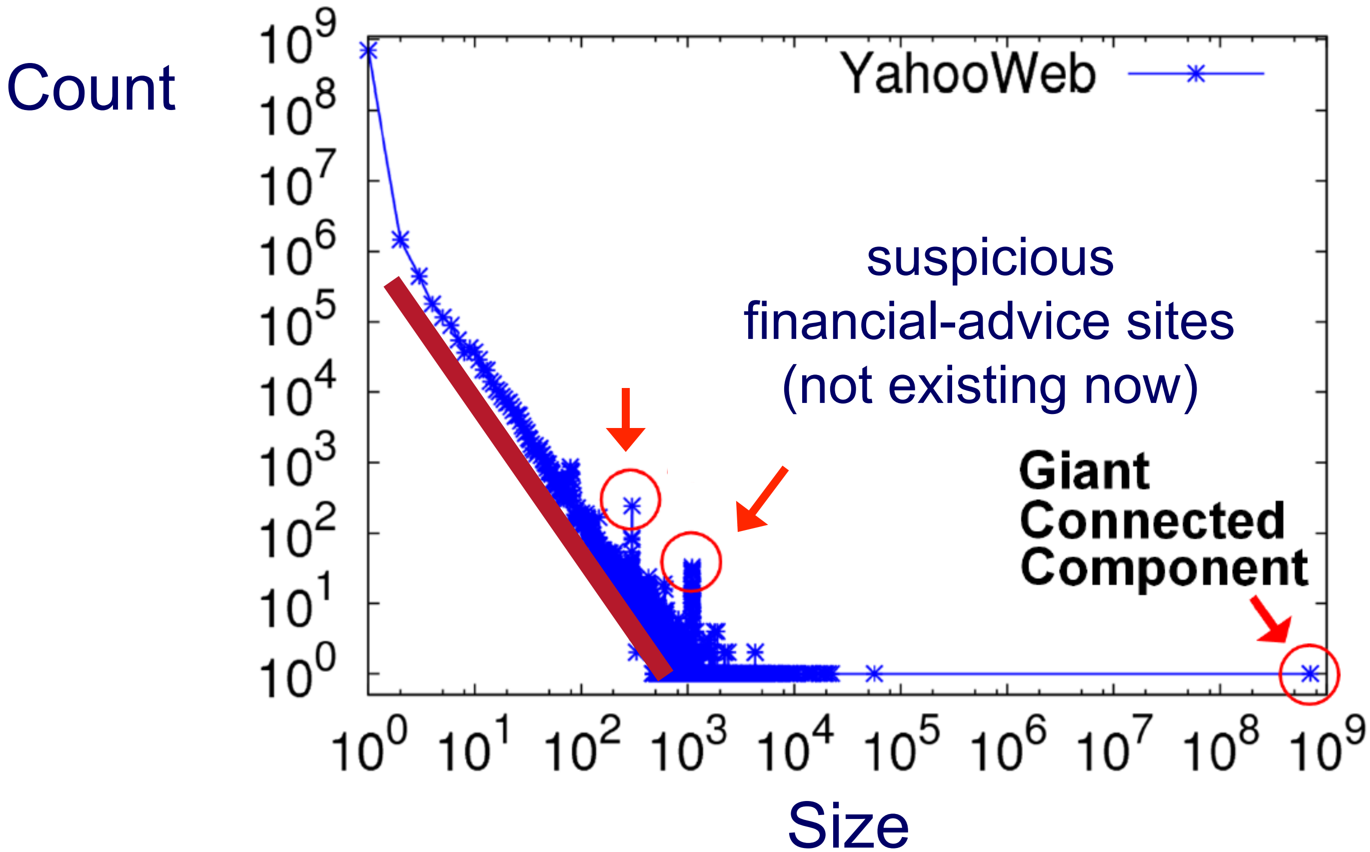


4) Spikes!

Size

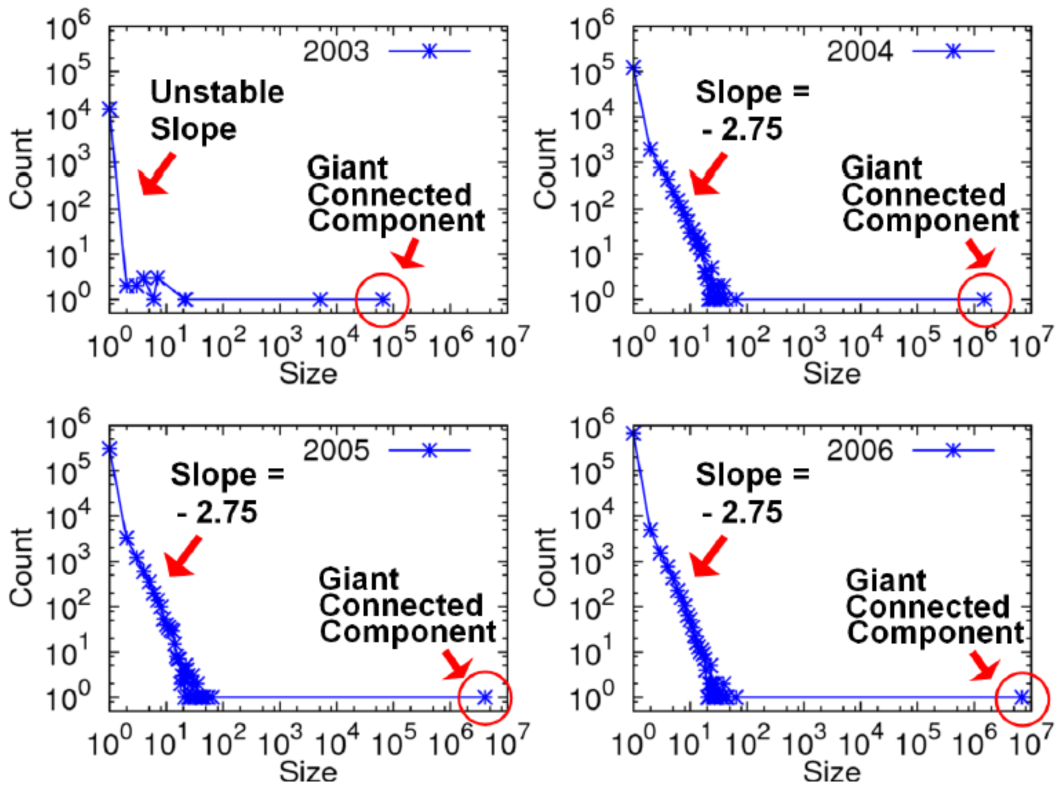
Example: GIM-V At Work

- Connected Components



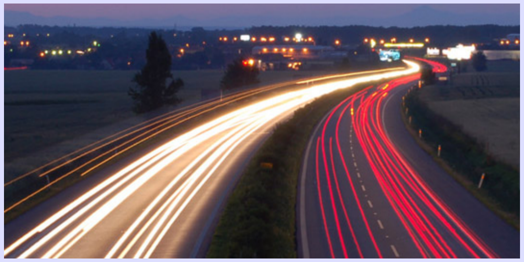
GIM-V At Work

- Connected Components over Time
- **LinkedIn: 7.5M nodes and 58M edges**



Stable tail slope after the gelling point

Roadmap



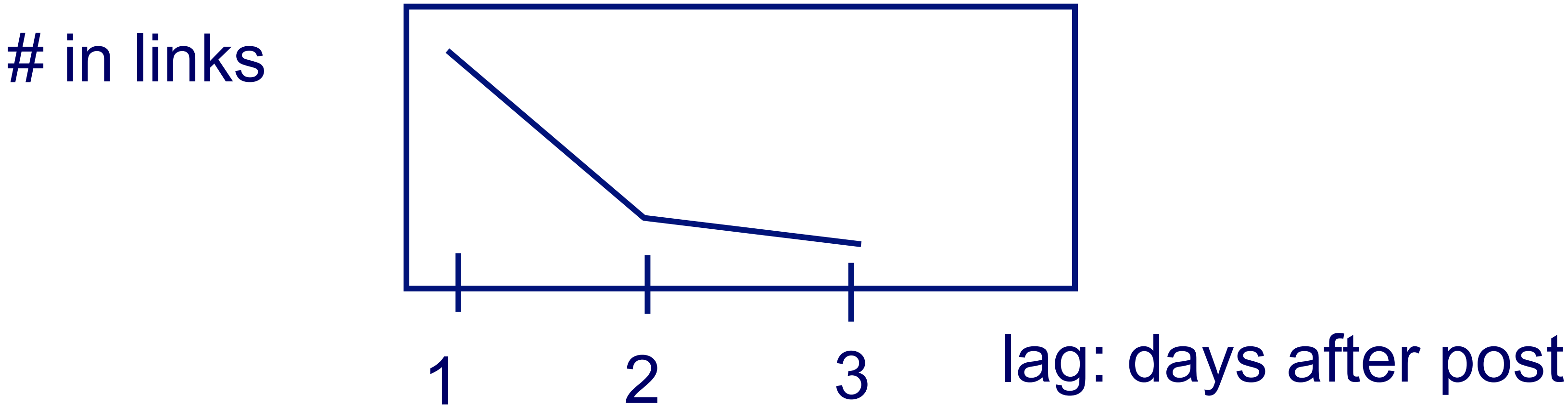
- Patterns in graphs
  - …
  - Time-evolving graphs
    - T1: shrinking diameter;
    - T2: densification
    - T3: connected components
    - T4: popularity over time
    - T5: phonecall patterns
- …

Timing for Blogs

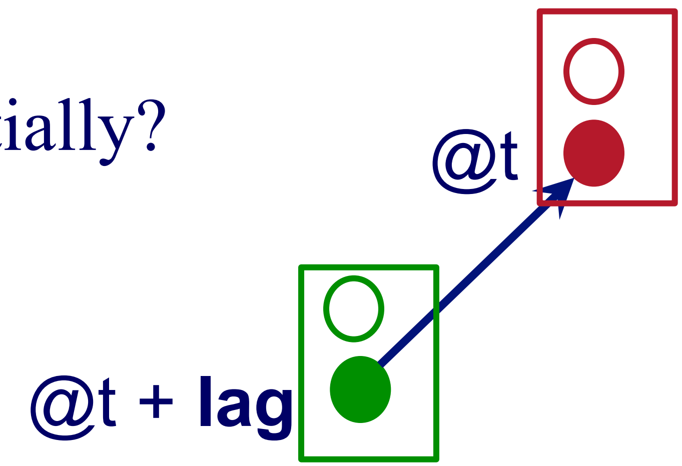
- with Mary McGlohon (CMU->Google)
- Jure Leskovec (CMU->Stanford)
- Natalie Glance (now at Google)
- Mat Hurst (now at MSR)

[SDM’07]

T.4 : popularity over time

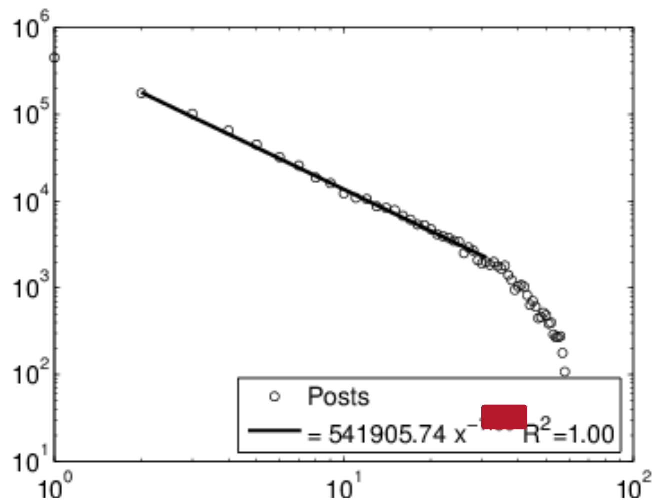


Post popularity drops-off – exponentially?



T.4 : popularity over time

in links
(**log**)

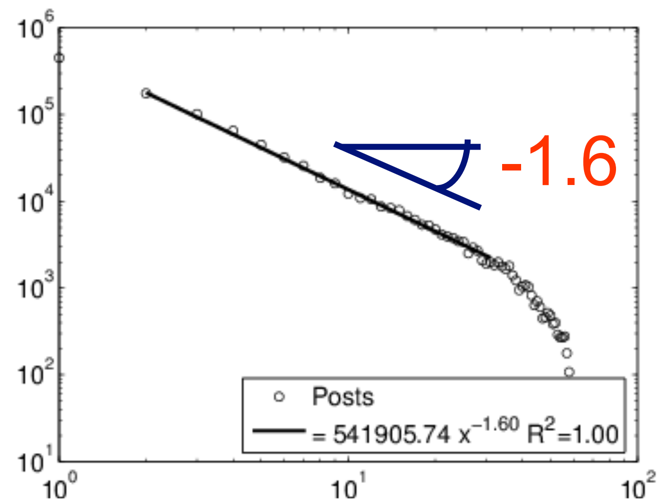


days after post
(**log**)

Post popularity drops-off – exponentially?
POWER LAW!
Exponent?

T.4 : popularity over time

in links (**log**)



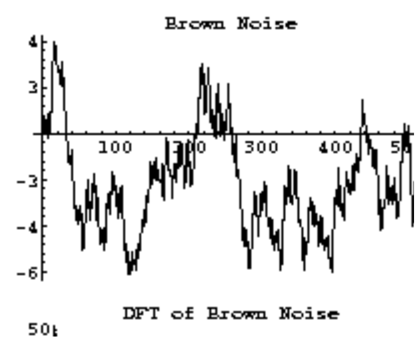
days after post (**log**)

Post popularity drops-off – ~~exponentially~~?

POWER LAW!

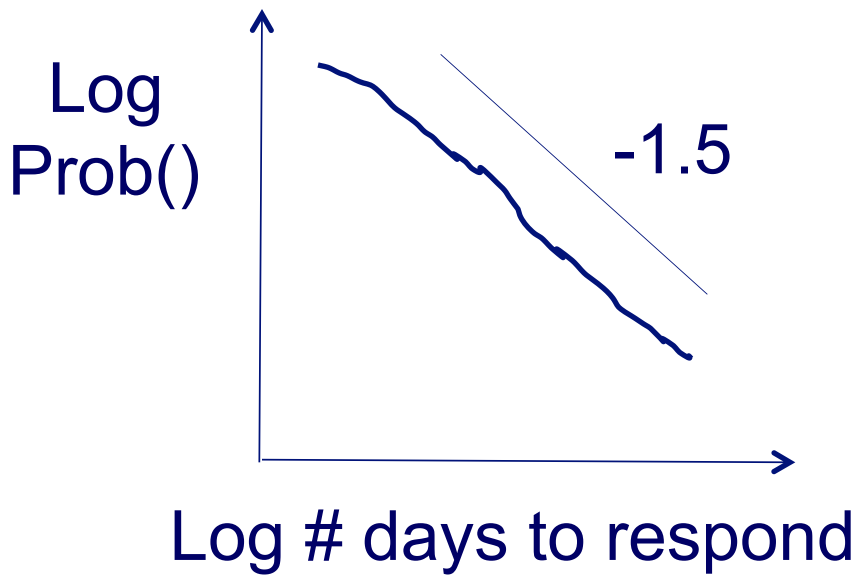
Exponent? -1.6

- close to -1.5: Barabasi's stack model
- and like the zero-crossings of a random walk

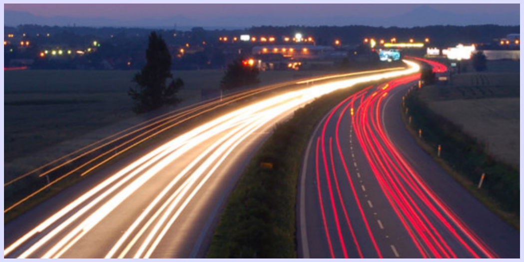


-1.5 slope

J. G. Oliveira & A.-L. Barabási Human Dynamics: The Correspondence Patterns of Darwin and Einstein. *Nature* **437,** 1251 (2005) . [PDF]



Roadmap



- Patterns in graphs
  - …
  - Time-evolving graphs
    - T1: shrinking diameter;
    - T2: densification
    - T3: connected components
    - T4: popularity over time
    - T5: phonecall patterns
- …

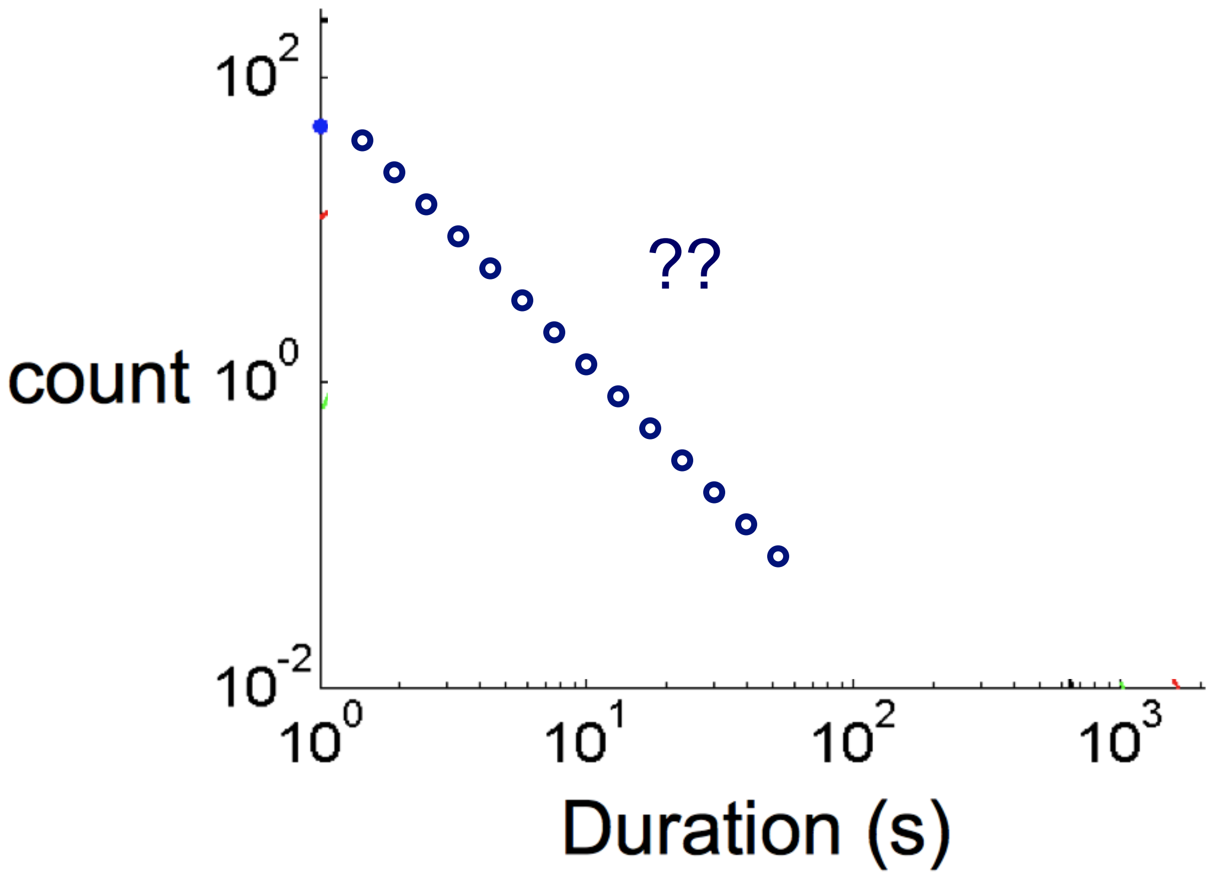
T.5: duration of phonecalls

Surprising Patterns for the Call Duration Distribution of Mobile Phone Users

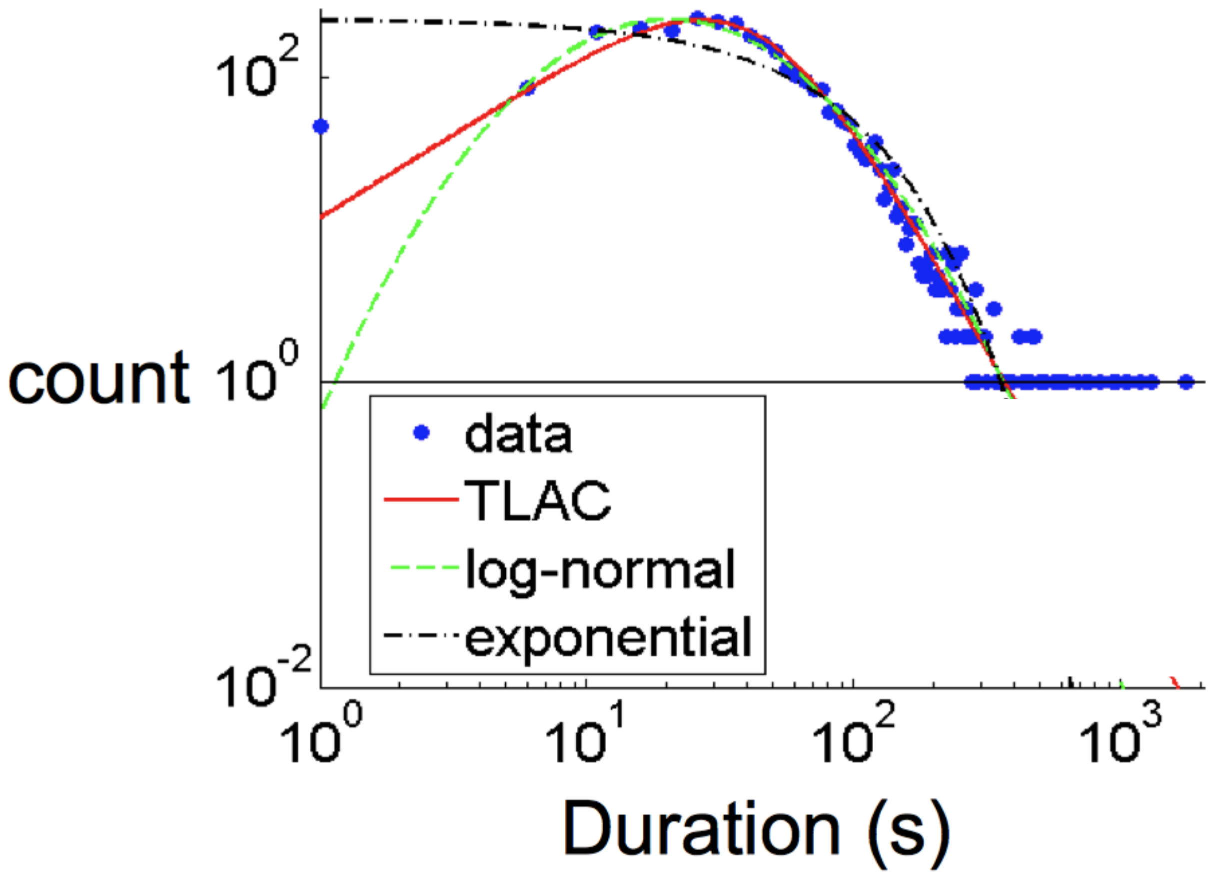
Pedro O. S. Vaz de Melo, Leman Akoglu, Christos Faloutsos, Antonio A. F. Loureiro

PKDD 2010

Probably, power law (?)

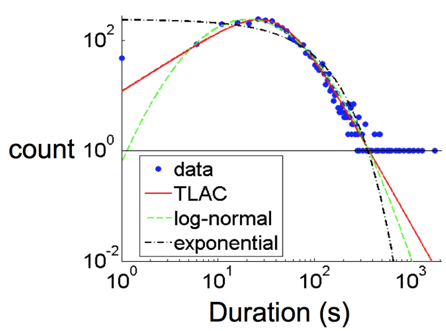


No Power Law!


count
10^2
10^0
10^{-2}
10^0
10^1
10^2
10^3
Duration (s)
data
TLAC
log-normal
exponential

‘TLaC: Lazy Contractor’

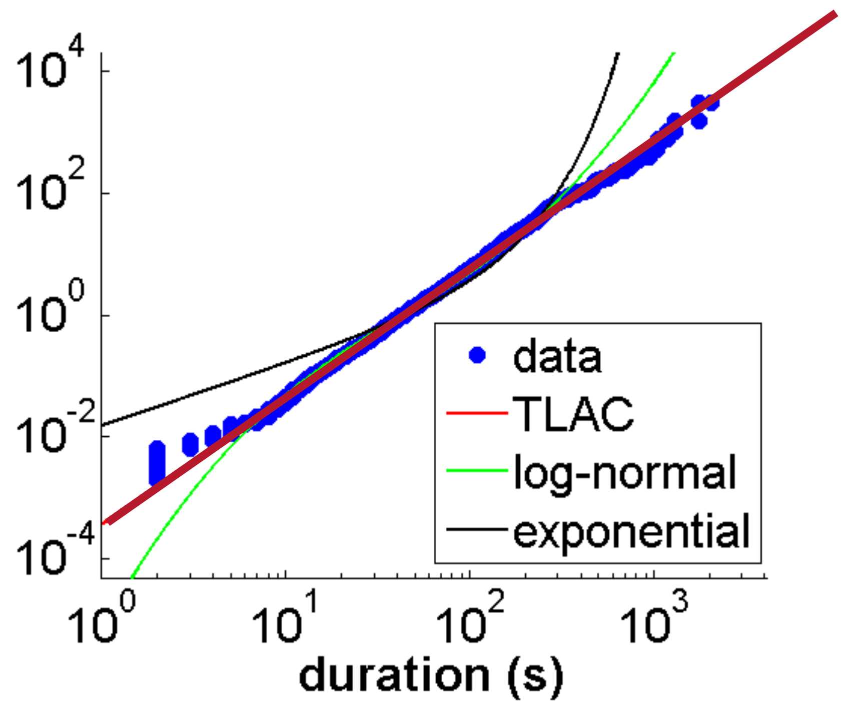
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

Casualties(<x): Survivors(>=x)

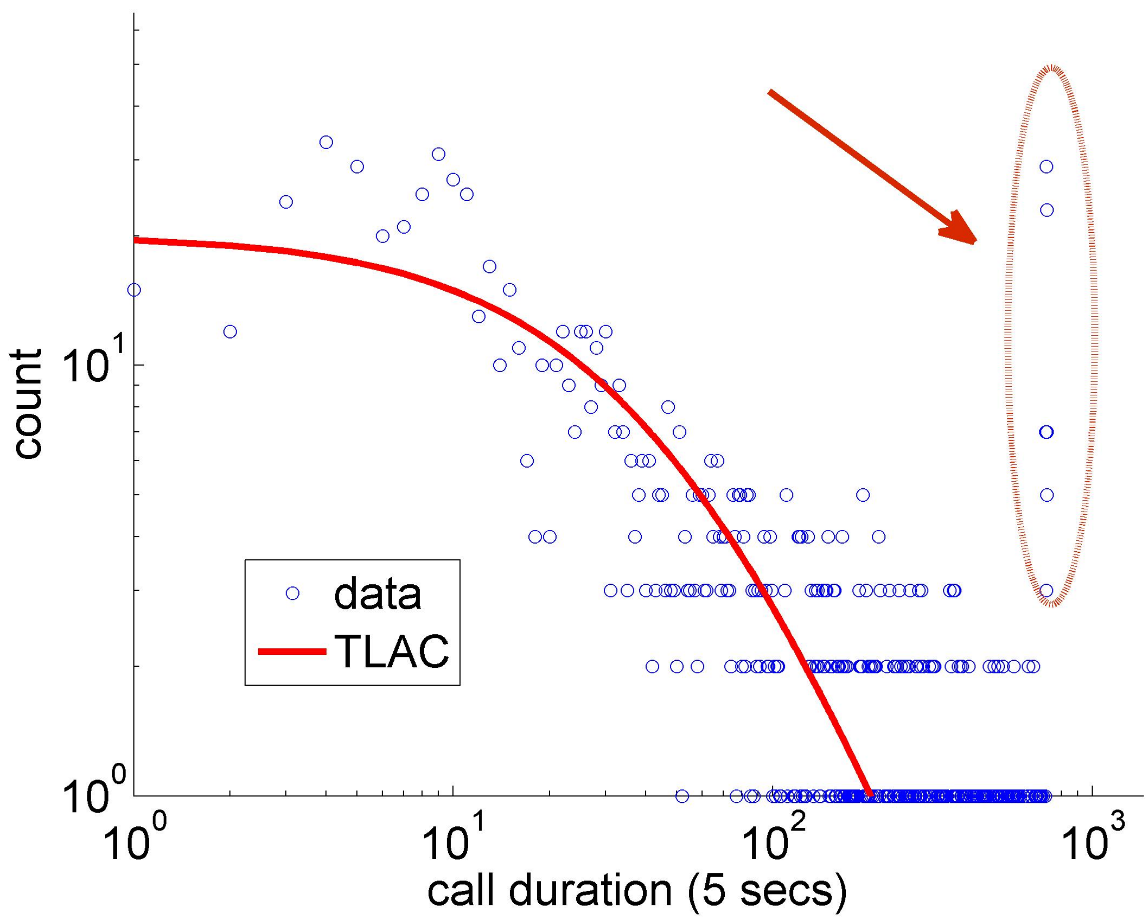
== power law



Data Description

- Data from a private mobile operator of a large city
  - 4 months of data
  - 3.1 million users
  - more than 1 billion phone records
- Over 96% of ‘talkative’ users obeyed a TLAC distribution (‘talkative’: >30 calls)

Outliers:



RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

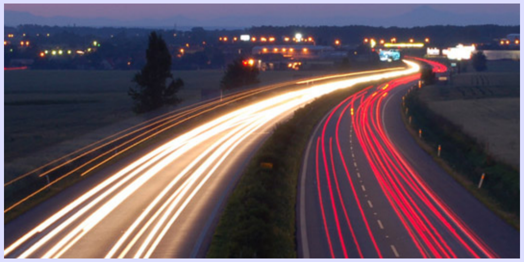
Carnegie Mellon

Real Graph Patterns

	unweighted	weighted
static	✓ **P01**. Power-law degree distribution [Faloutsos et. al.`99, Kleinberg et. al.`99, Chakrabarti et. al. `04, Newman`04] ✓ **P02**. Triangle Power Law [Tsourakakis `08] ✓ **P03**. Eigenvalue Power Law [Siganos et. al. `03] **P04**. Community structure [Flake et. al.`02, Girvan and Newman `02] ✓ **P05**. Clique Power Laws [Du et. al. ‘09]	**P12**. Snapshot Power Law [McGlohon et. al. `08]
dynamic	✓ **P06**. Densification Power Law [Leskovec et. al.`05] ✓ **P07**. Small and shrinking diameter [Albert and Barabási `99, Leskovec et. al. ‘05, McGlohon et. al. ‘08] **P08**. Gelling point [McGlohon et. al. `08] ✓ **P09**. Constant size 2nd and 3rd connected components [McGlohon et. al. `08] **P10**. Principal Eigenvalue Power Law [Akoglu et. al. `08] **P11**. Bursty/self-similar edge/weight additions [Gomez and Santonja `98, Gribble et. al. `98, Crovella and Bestavros `99, McGlohon et .al. `08]	✓ **P13**. Weight Power Law [McGlohon et. al. `08] ✓ **P14**. Skewed call duration distributions [Vaz de Melo et. al. `10]

RTG: A Recursive Realistic Graph Generator using Random Typing
Leman Akoglu and Christos Faloutsos. *ECML PKDD*’09.

Roadmap



- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

OddBall: Spotting Anomalies in Weighted Graphs

Leman Akoglu, Mary McGlohon, Christos Faloutsos

Carnegie Mellon University

School of Computer Science

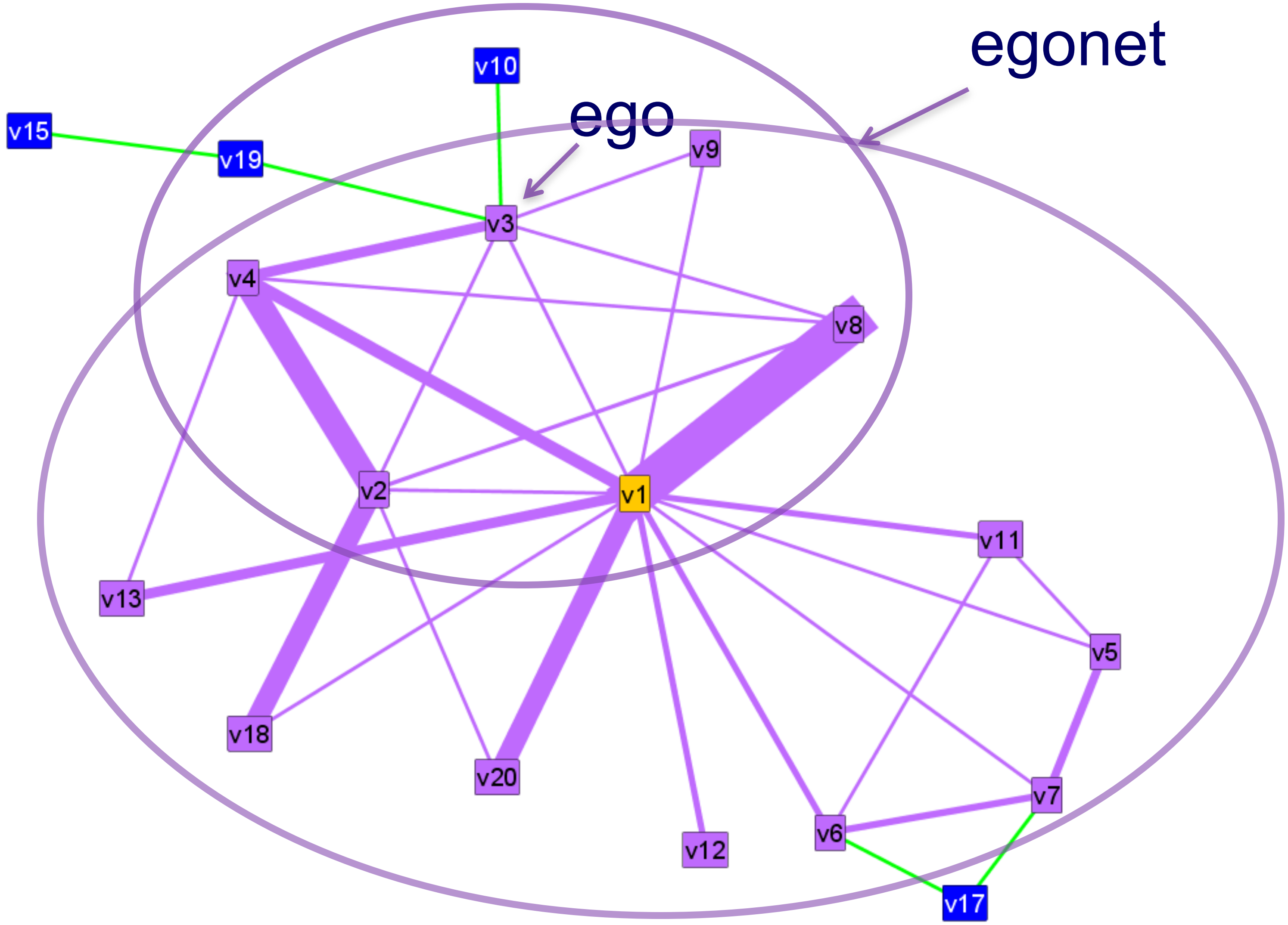
PAKDD 2010, Hyderabad, India

Main idea

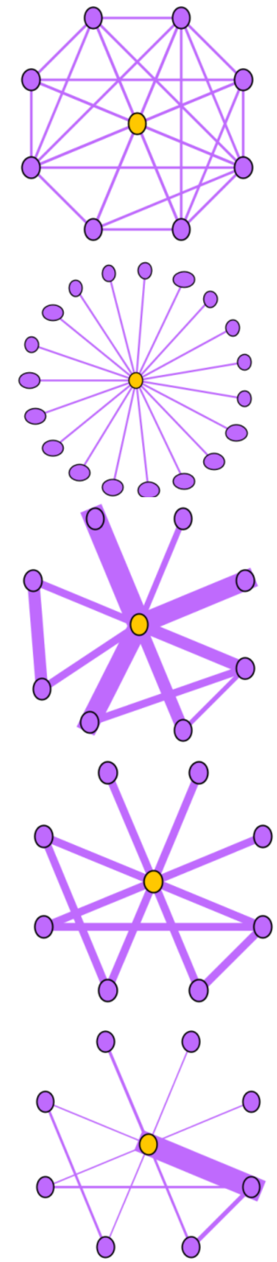
For each node,

- extract 'ego-net' (=1-step-away neighbors)
- Extract features (#edges, total weight, etc etc)
- Compare with the rest of the population

What is an egonet?

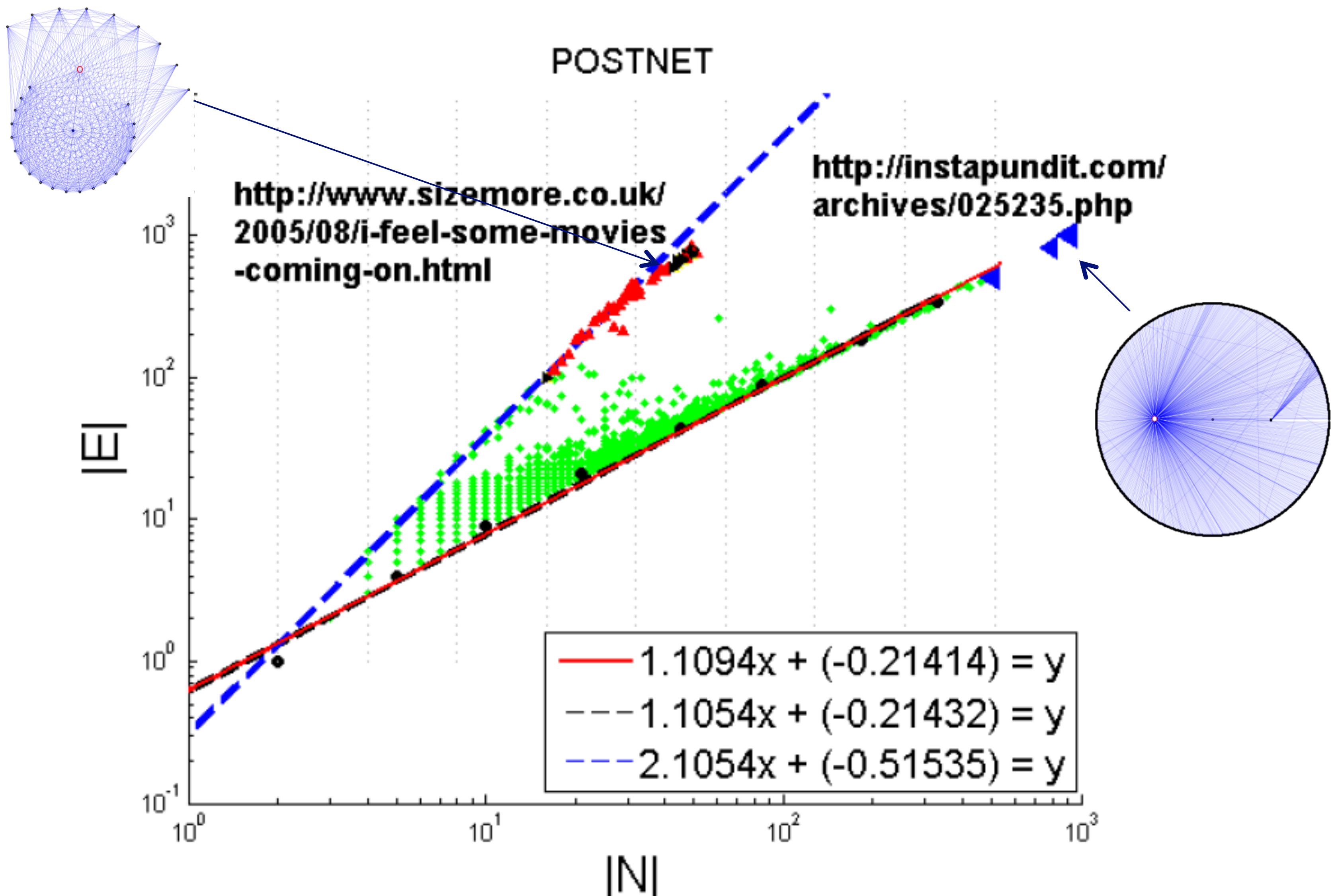


Selected Features

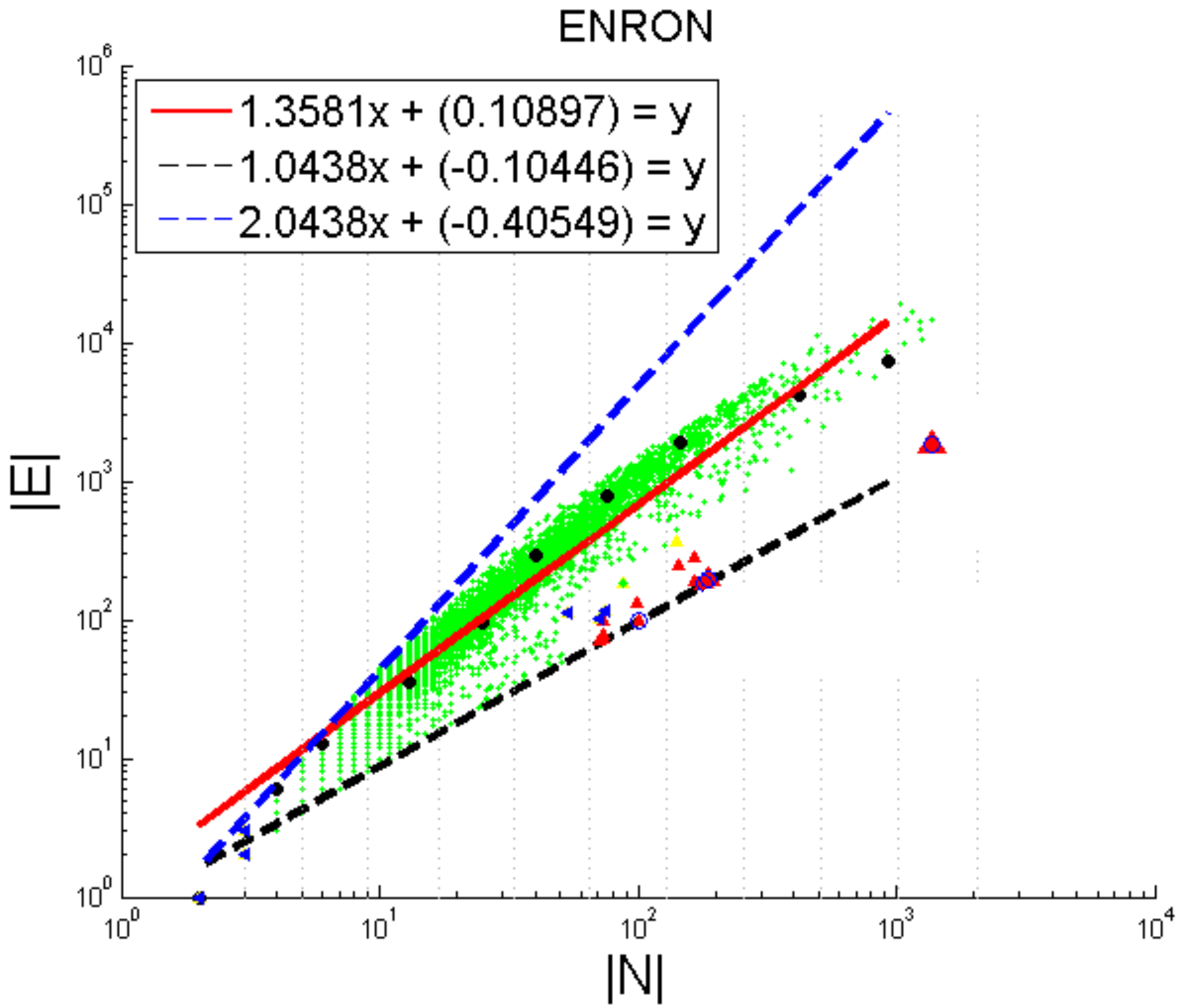


- N_i: number of neighbors (degree) of ego i
- E_i: number of edges in egonet i
- W_i: total weight of egonet i
- $\lambda_{w,i}$: principal eigenvalue of the weighted adjacency matrix of egonet I

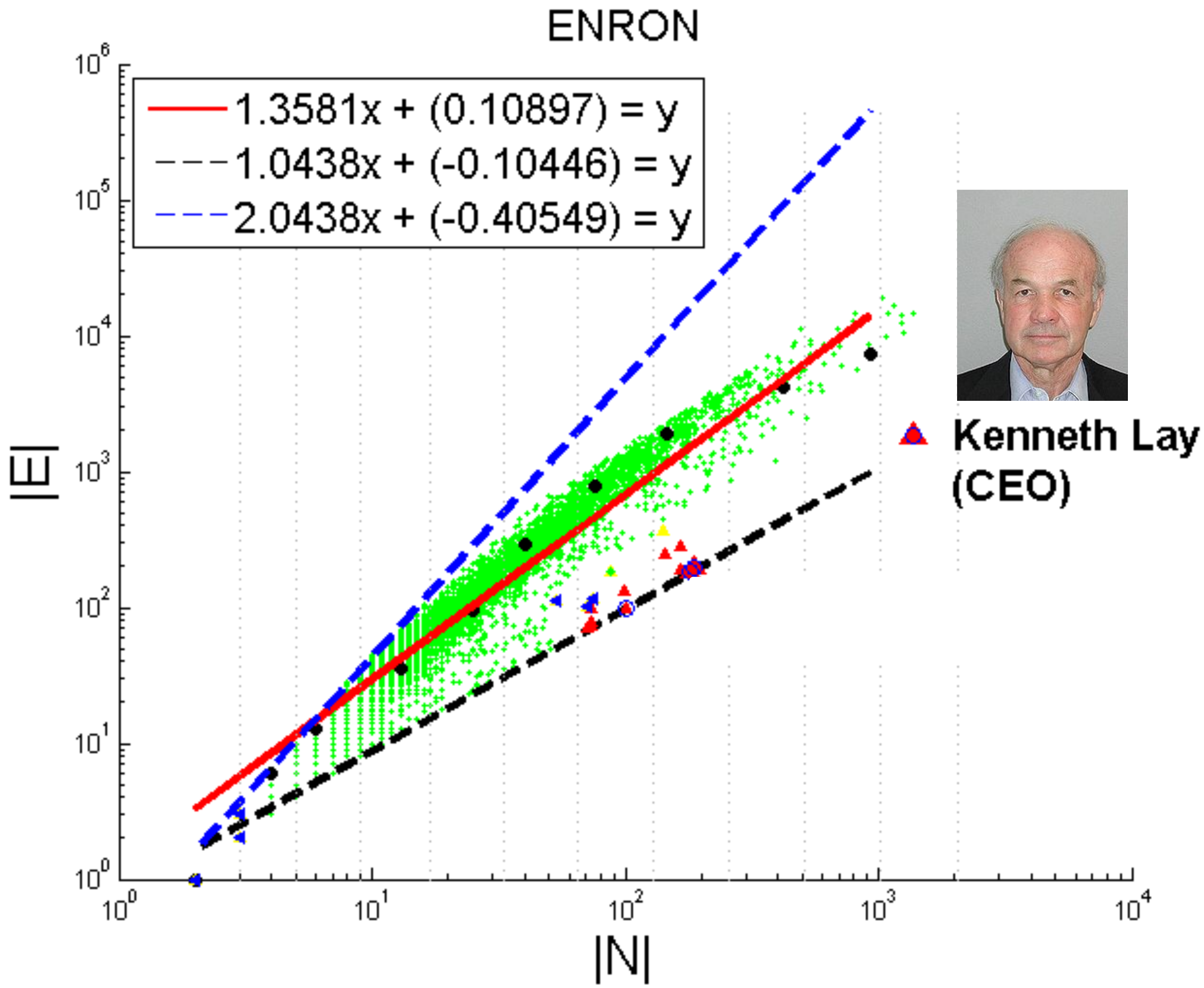
Near-Clique/Star



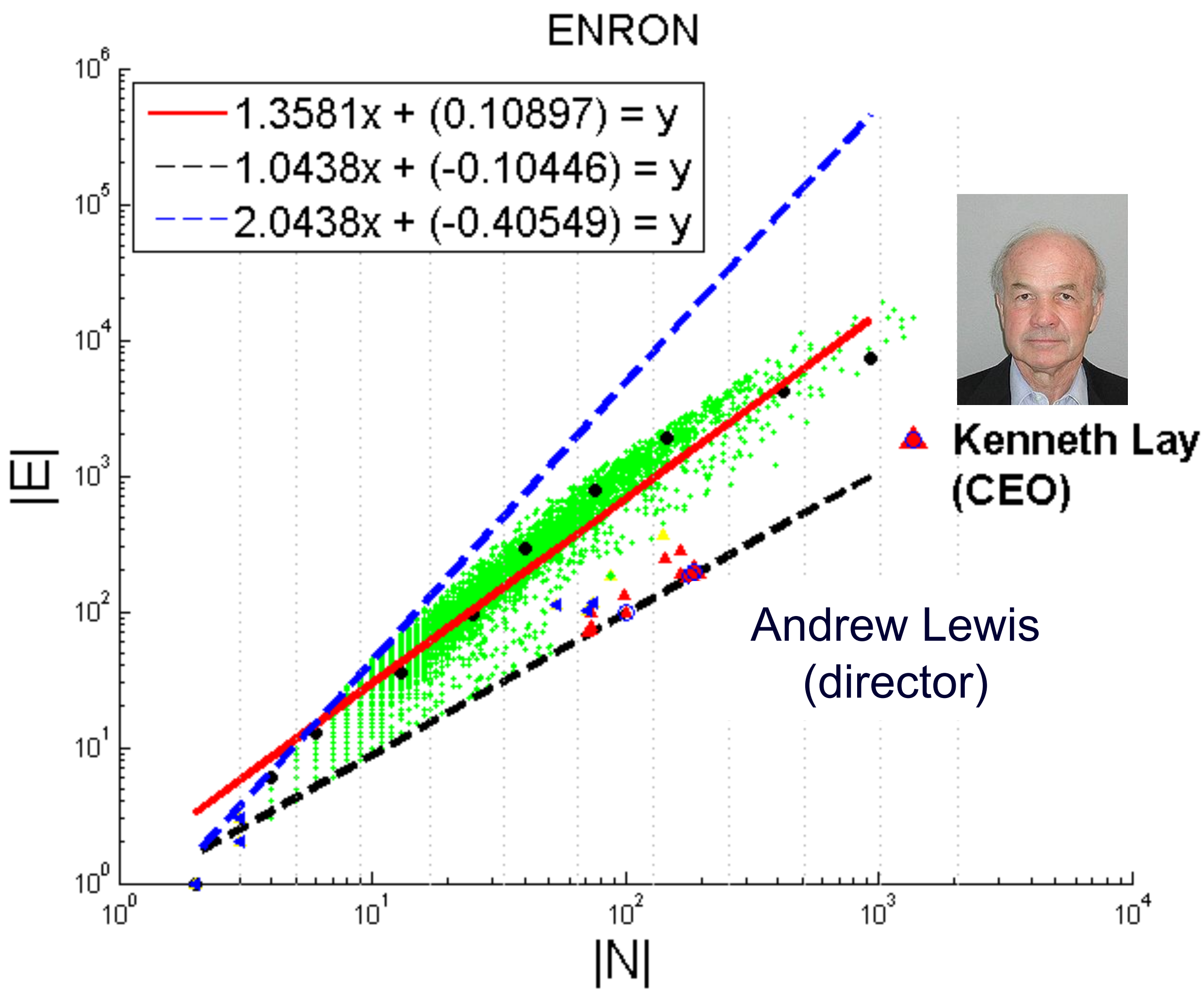
Near-Clique/Star



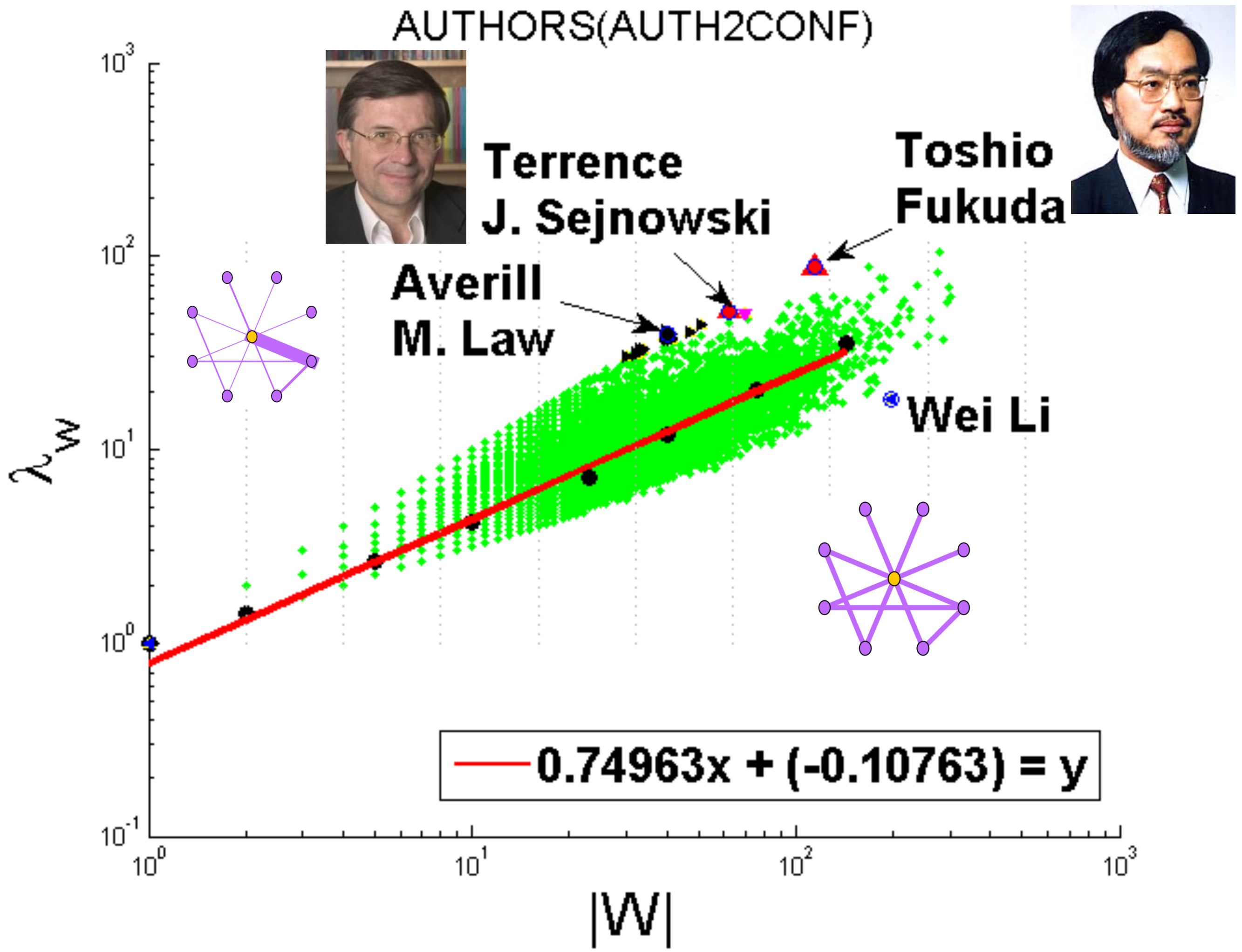
Near-Clique/Star



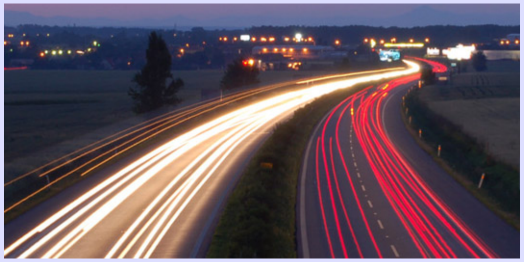
Near-Clique/Star



Dominant Heavy Link



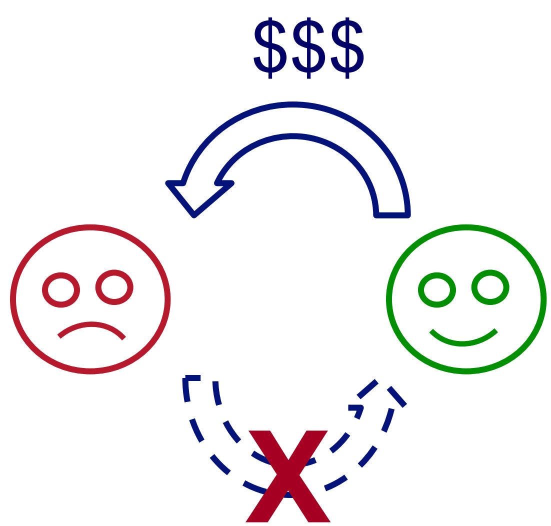
Roadmap



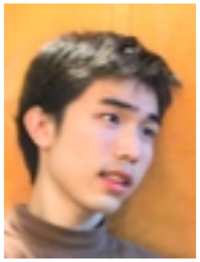
- Patterns in graphs
  - overview
  - Static graphs
  - Weighted graphs
  - Time-evolving graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

NetProbe: The Problem

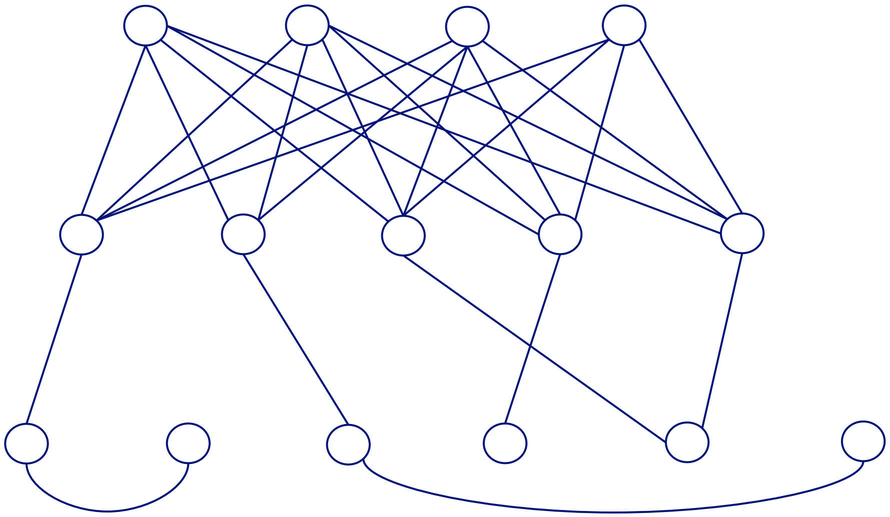
Find **bad sellers (fraudsters)** on eBay who don’t deliver their (expensive) items



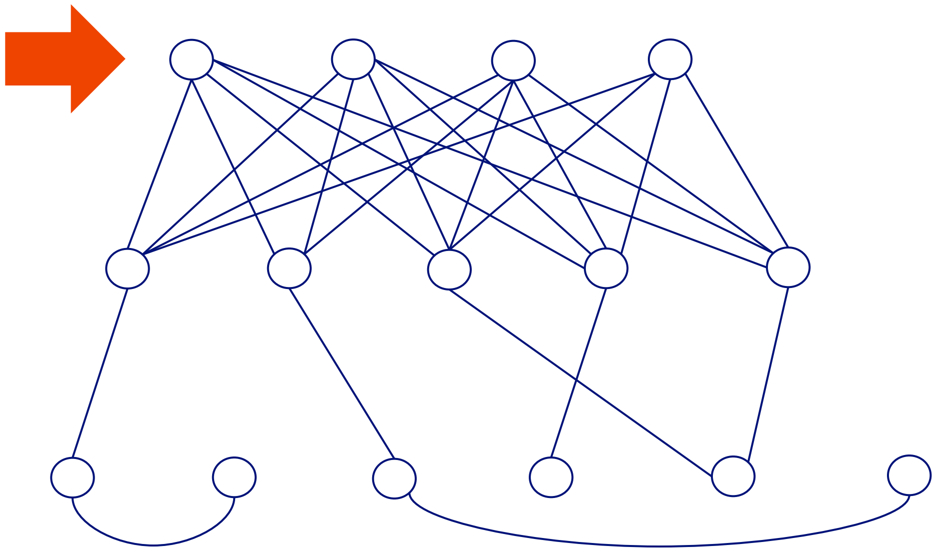
E-bay Fraud detection



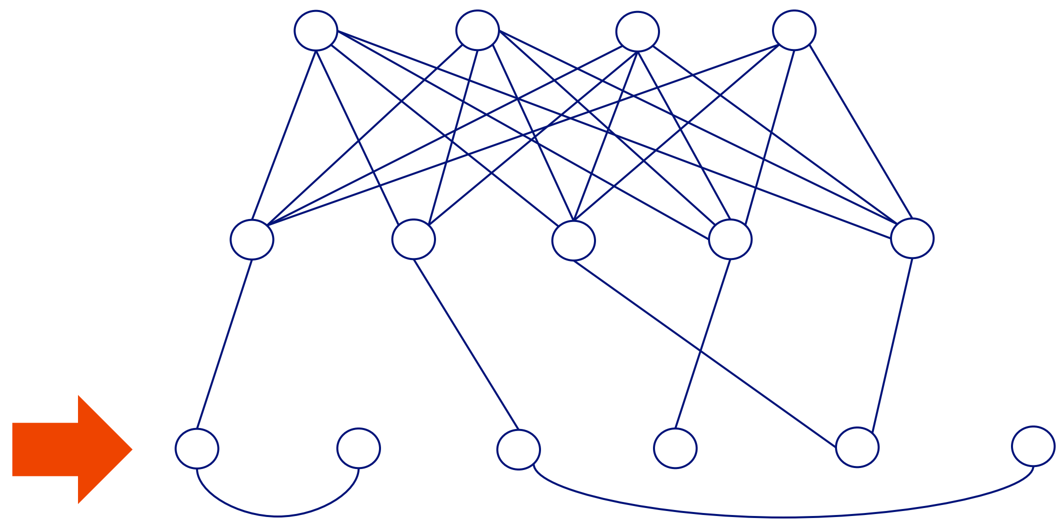
w/ Polo Chau &
Shashank Pandit, CMU
[www'07]



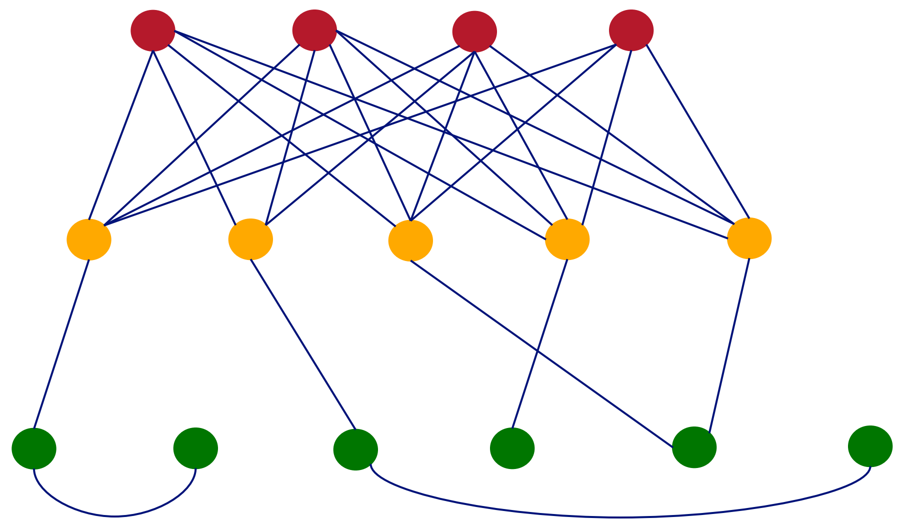
E-bay Fraud detection



E-bay Fraud detection

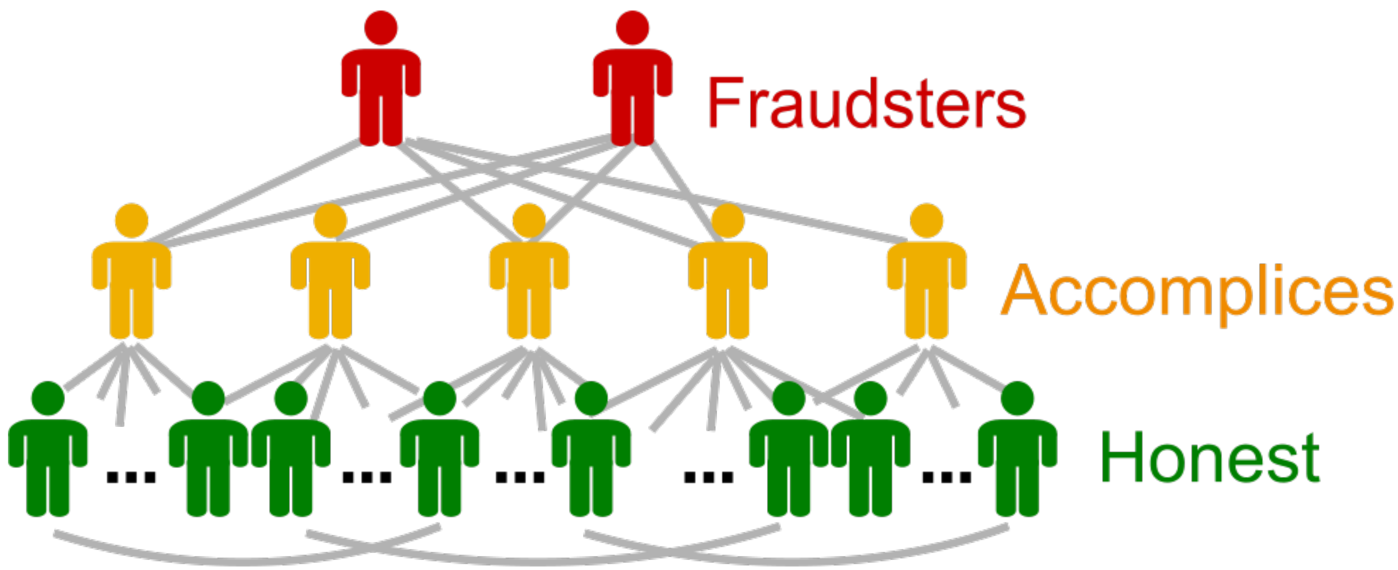


E-bay Fraud detection - NetProbe



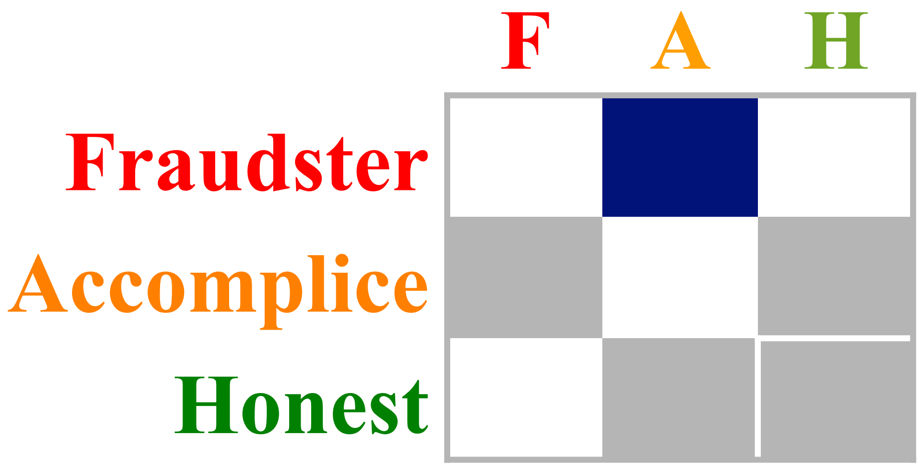
NetProbe: Key Ideas

- Fraudsters **fabricate their reputation** by "trading" with their accomplices
- Transactions form **near bipartite cores**
- How to detect them?

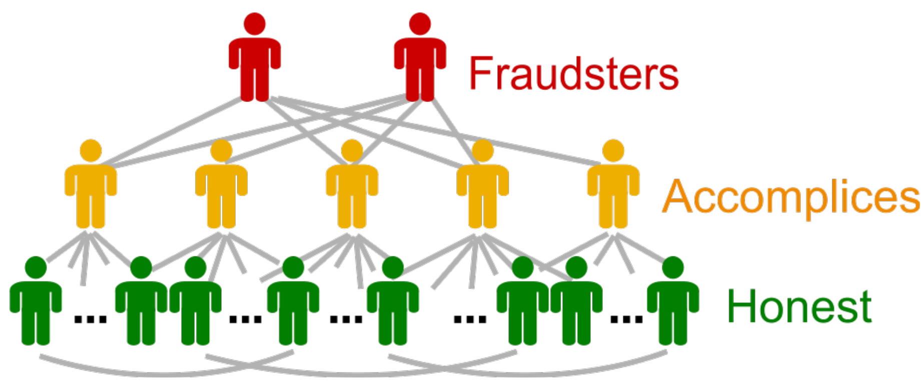


NetProbe: Key Ideas

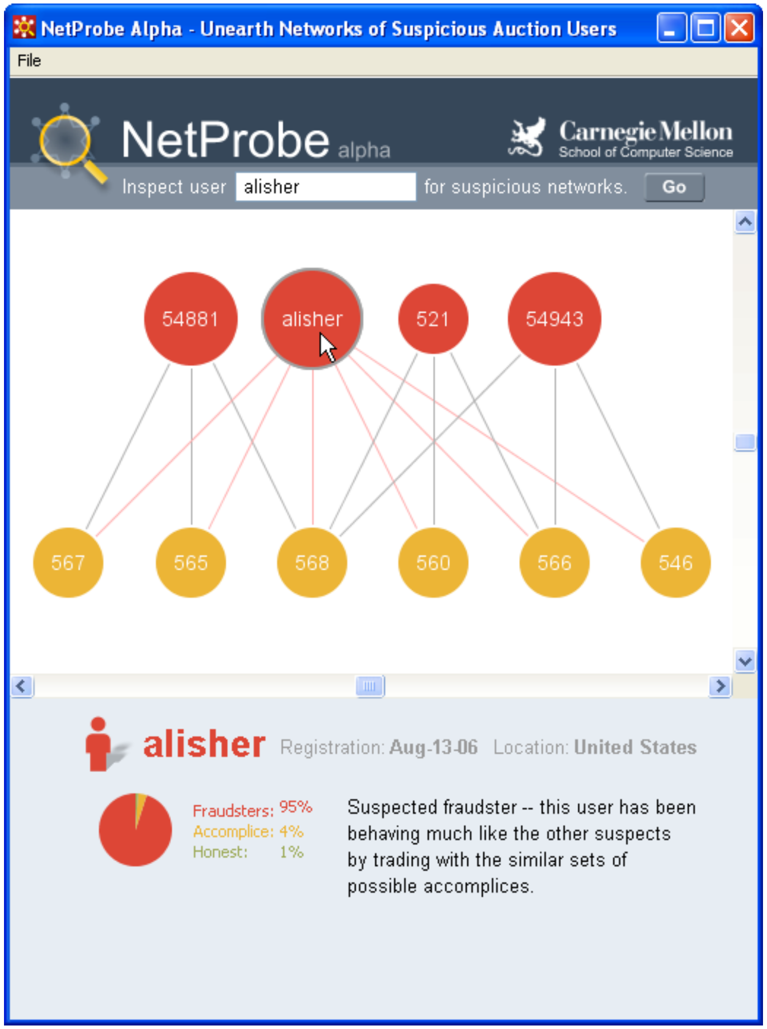
Use ‘**Belief Propagation**’ and ~**heterophily**

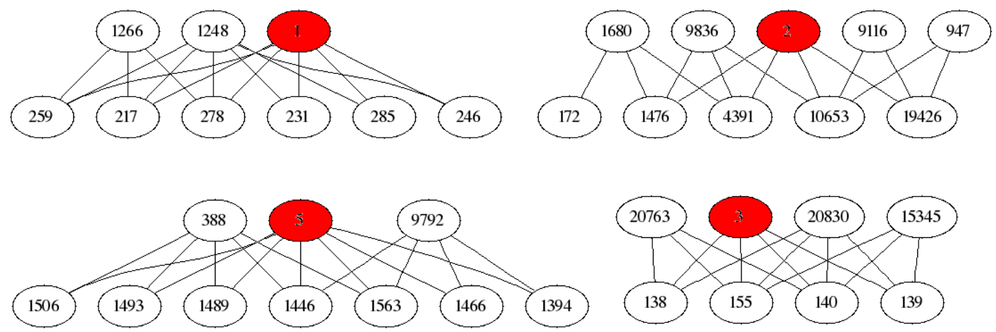


Darker means more likely

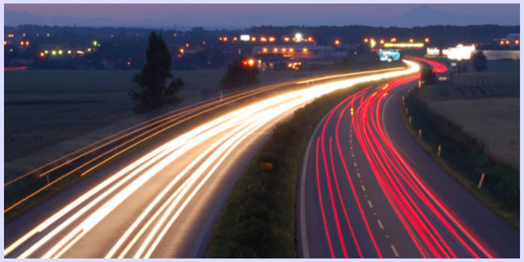


NetProbe: Main Results





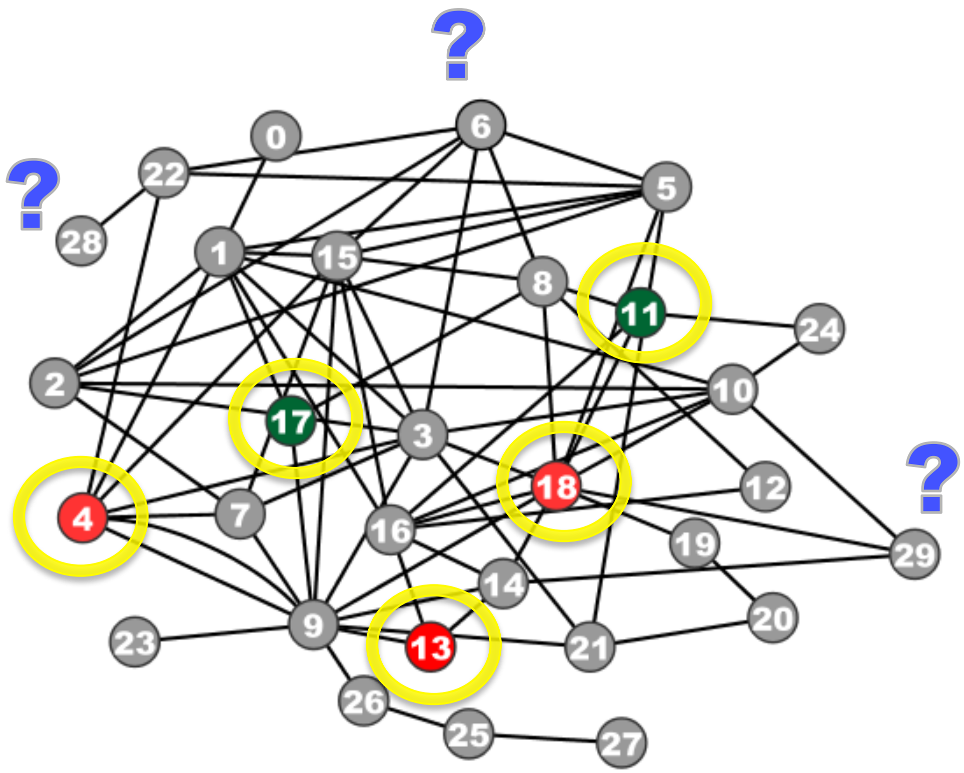
Roadmap



- Patterns in graphs
- Anomaly Detection
- Application: ebay fraud
  - How-to: Belief Propagation
- Conclusions

Guilt-by-Association Techniques

details

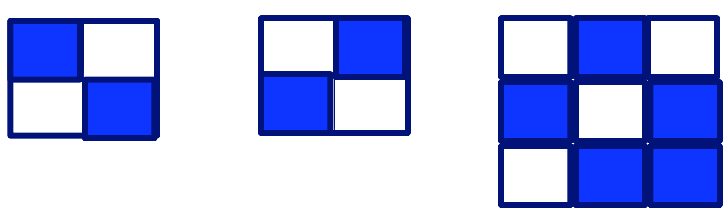


Given:
- graph and
- few labeled nodes

Find: class (red/green) for rest nodes

Assuming: network effects (homophily/ heterophily, etc)

red
green

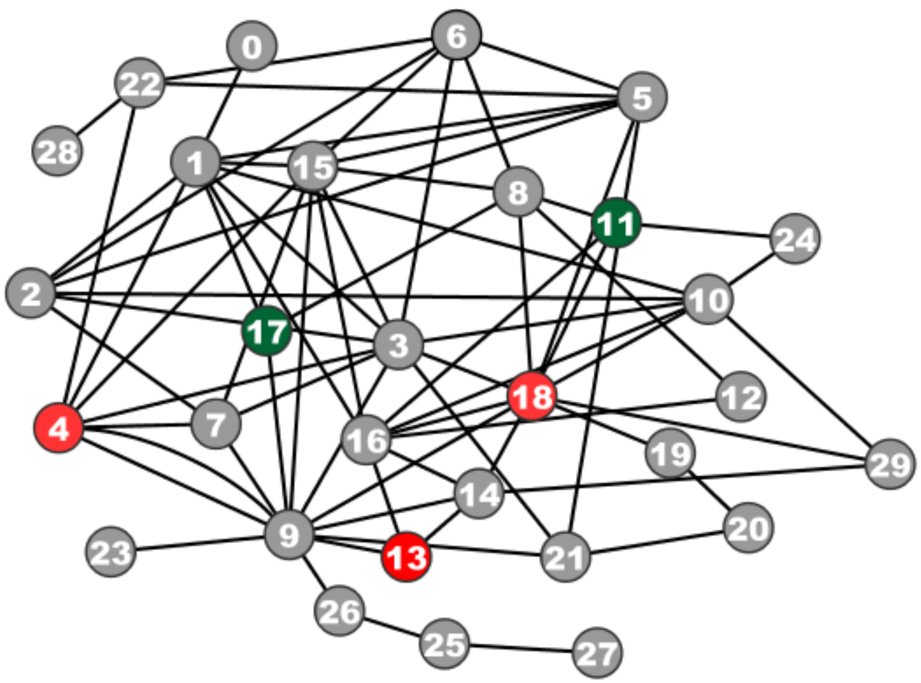


F
A
H

details

Correspondence of Methods

Random Walk with Restarts (**RWR**) Google
Semi-supervised Learning (**SSL**)
Belief Propagation (**BP**) Bayesian

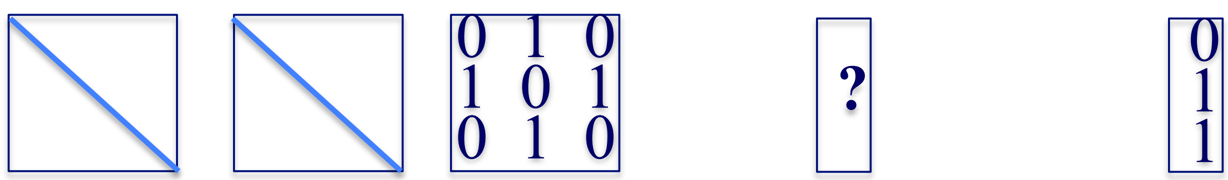


details

Correspondence of Methods

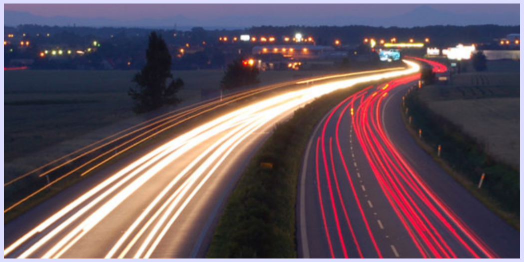
Random Walk with Restarts (**RWR**) $\approx$
Semi-supervised Learning (**SSL**) $\approx$
Belief Propagation (**BP**)

Method	Matrix		unknown		known
RWR	$[\mathbf{I} - c\ \mathbf{A}\mathbf{D}^{-1}]$	$\times$	$\mathbf{x}$	$=$	$(1-c)\mathbf{y}$
SSL	$[\mathbf{I} + \alpha(\mathbf{D} - \mathbf{A})]$	$\times$	$\mathbf{x}$	$=$	$\mathbf{y}$
FABP	$[\mathbf{I} + a\ \mathbf{D} - c'\mathbf{A}]$	$\times$	$\mathbf{b_h}$	$=$	$\boldsymbol{\phi}_\mathbf{h}$



Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. Danai Koutra, et al *PKDD'11*

Roadmap

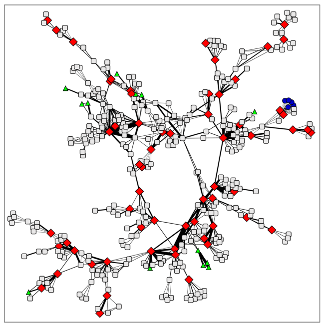


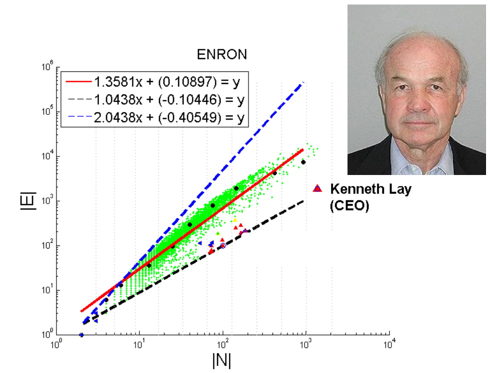
- Patterns in graphs
- Anomaly Detection
- Application: ebay fraud
- Conclusions

Overall conclusions

- Roles:
  - Past work in social networks (‘regular’, ‘structural’ etc)
  - Scalable algo’s to find such roles
- Anomalies & patterns
  - Static (power-laws, ‘six degrees’)
  - Weighted (super-linearity)
  - Time-evolving (densification, -1.5 exponent)

OVERALL CONCLUSIONS – high level:





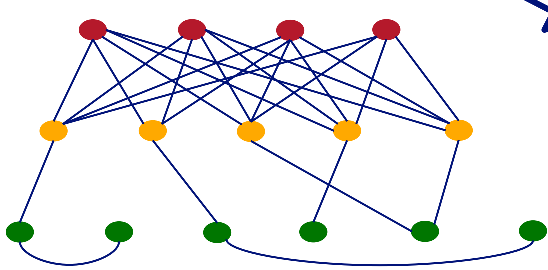
Features

Roles

Anomalies

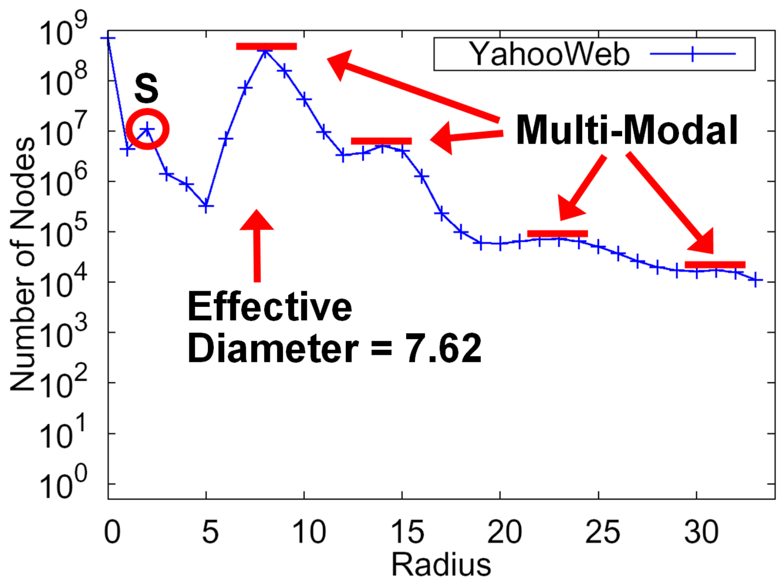
= rare roles

Patterns



OVERALL CONCLUSIONS – high level

- **BIG DATA: ->** roles/patterns/outliers that are invisible otherwise



References

- Leman Akoglu, Christos Faloutsos: *RTG: A Recursive Realistic Graph Generator Using Random Typing*. ECML/PKDD (1) 2009: 13-28

- Deepayan Chakrabarti, Christos Faloutsos: *Graph mining: Laws, generators, and algorithms*. ACM Comput. Surv. 38(1): (2006)

References

- Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jure Leskovec, Christos Faloutsos: *Epidemic thresholds in real networks.* ACM Trans. Inf. Syst. Secur. 10(4): (2008)

- Deepayan Chakrabarti, Jure Leskovec, Christos Faloutsos, Samuel Madden, Carlos Guestrin, Michalis Faloutsos: *Information Survival Threshold in Sensor and P2P Networks*. INFOCOM 2007: 1316-1324

References

- Christos Faloutsos, Tamara G. Kolda, Jimeng Sun: *Mining large graphs and streams using matrix and tensor tools*. Tutorial, SIGMOD Conference 2007: 1174

References

- T. G. Kolda and J. Sun. *Scalable Tensor Decompositions for Multi-aspect Data Mining*. In: ICDM 2008, pp. 363-372, December 2008.

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

Carnegie Mellon

References

- Jure Leskovec, Jon Kleinberg and Christos Faloutsos *Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations*, KDD 2005 (Best Research paper award).
- Jure Leskovec, Deepayan Chakrabarti, Jon M. Kleinberg, Christos Faloutsos: *Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication*. PKDD 2005: 133-145

References

- Jimeng Sun, Yinglian Xie, Hui Zhang, Christos Faloutsos. *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM, Minneapolis, Minnesota, Apr 2007.
- Jimeng Sun, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos, *GraphScope: Parameter-free Mining of Large Time-evolving Graphs* ACM SIGKDD Conference, San Jose, CA, August 2007

References

- Jimeng Sun, Dacheng Tao, Christos Faloutsos: *Beyond streams and graphs: dynamic tensor analysis*. KDD 2006: 374-383

References

- Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan, *Fast Random Walk with Restart and Its Applications*, ICDM 2006, Hong Kong.
- Hanghang Tong, Christos Faloutsos, *Center-Piece Subgraphs: Problem Definition and Fast Solutions*, KDD 2006, Philadelphia, PA

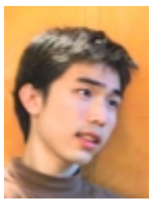
References

- Hanghang Tong, Christos Faloutsos, Brian Gallagher, Tina Eliassi-Rad: Fast best-effort pattern matching in large attributed graphs. KDD 2007: 737-746

Project info

www.cs.cmu.edu/~pegasus

Akoglu, Leman



Chau, Polo

Kang, U

Koutra, Danai

McGlohon, Mary

Prakash, Aditya

Tong, Hanghang

Thanks to: NSF IIS-0705359, IIS-0534205, CTA-INARC; ADAMS-DARPA; Yahoo (M45), LLNL, IBM, SPRINT, Google, INTEL, HP, iLab