

# Mining Billion-node Graphs: Patterns, Generators and Tools

*Christos Faloutsos*

CMU

(on sabbatical at google)

# Thank you!



- Jennifer



- Marianne

## Our goal:

Open source system for mining huge graphs:

PEGASUS project (PEta GrAph mining System)

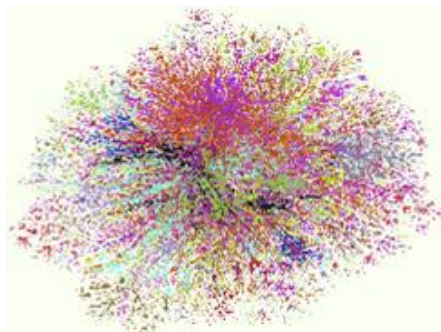
- [www.cs.cmu.edu/~pegasus](http://www.cs.cmu.edu/~pegasus)
- code and papers



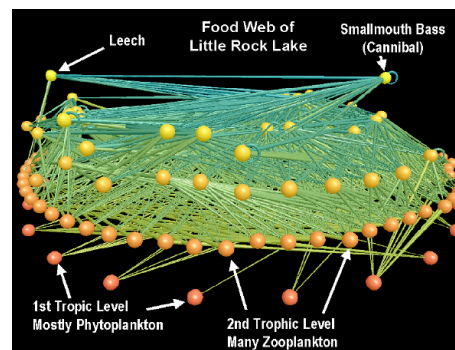
# Outline

- ➔ • Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
- Problem#3: Scalability
- Conclusions

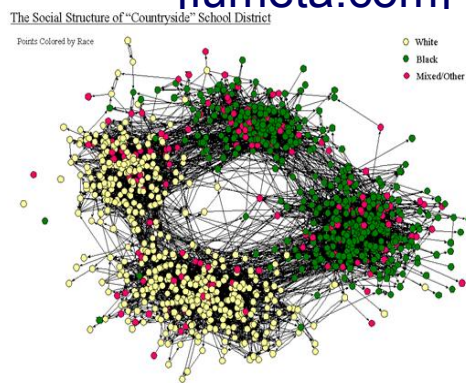
# Graphs - why should we care?



Internet Map  
[lumeta.com]



Food Web  
[Martinez '91]

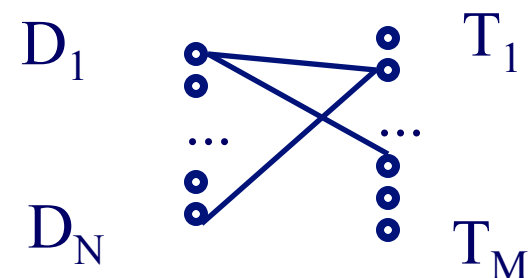


Friendship Network  
[Moody '01]

- Social networks
  - (facebook, orkut, ...)
- twitter

# Graphs - why should we care?

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



- web: hyper-text graph

- ... and more:

# Graphs - why should we care?

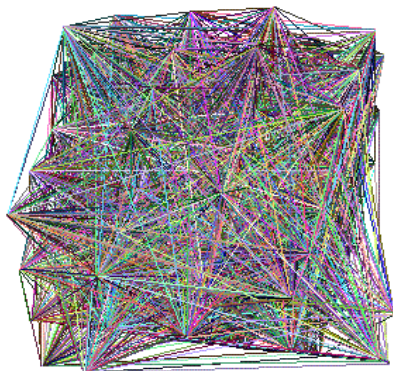
- ‘viral’ marketing
- web-log (‘blog’) news propagation
- computer network security: email/IP traffic and anomaly detection
- ....

# Outline

- Introduction – Motivation
- ➔ • Problem#1: Patterns in graphs
  - Static graphs
  - Weighted graphs
  - Time evolving graphs
- Problem#2: Tools
- Problem#3: Scalability
- Conclusions

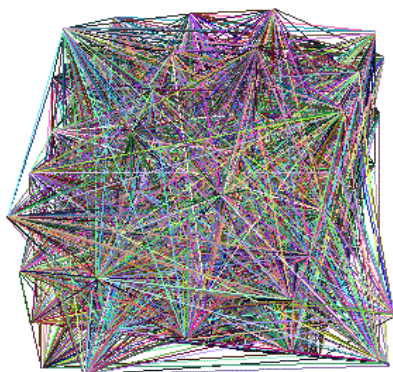


# Problem #1 - network and graph mining

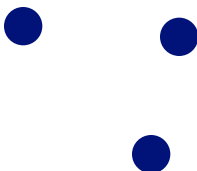


- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?

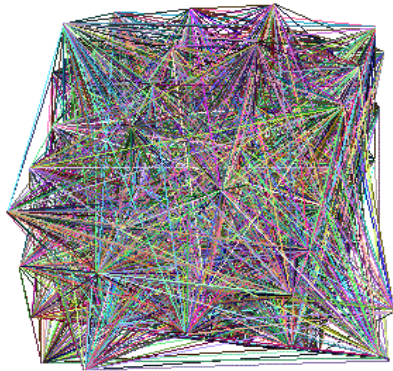
# Problem #1 - network and graph mining



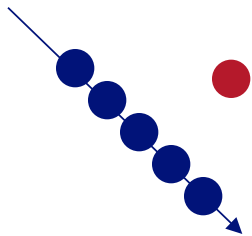
- What does the Internet look like?
- What does FaceBook look like?
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**



# 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?

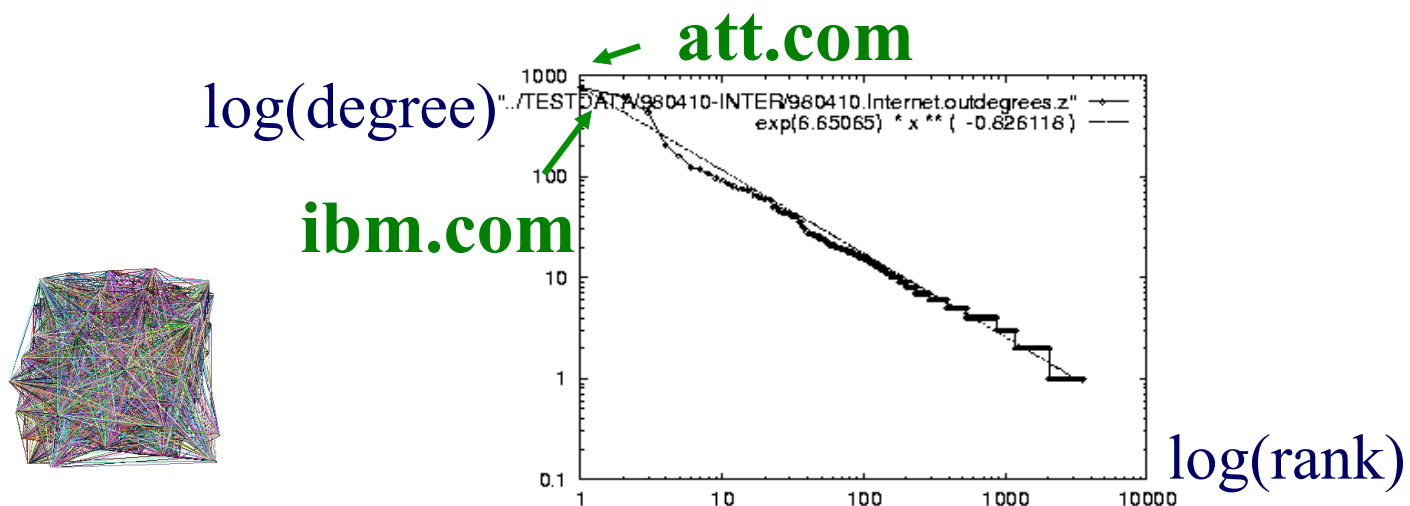
# Laws and patterns

- Are real graphs random?
- A: NO!!
  - Diameter
  - in- and out- degree distributions
  - other (surprising) patterns
- So, let's look at the data

# Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

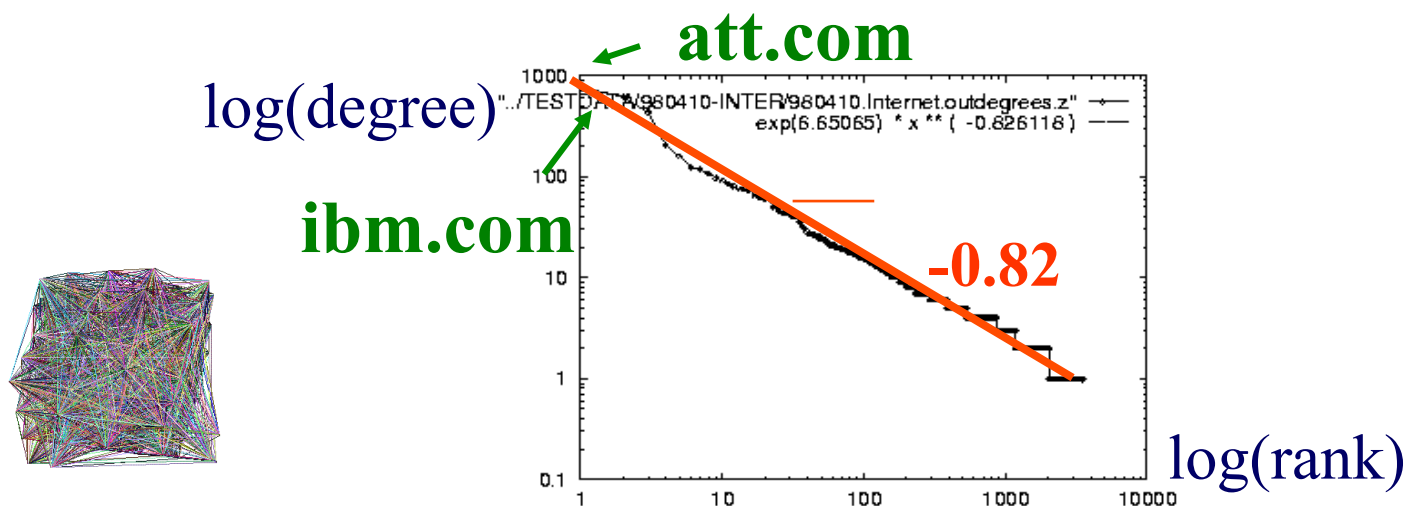
internet domains



# Solution# S.1

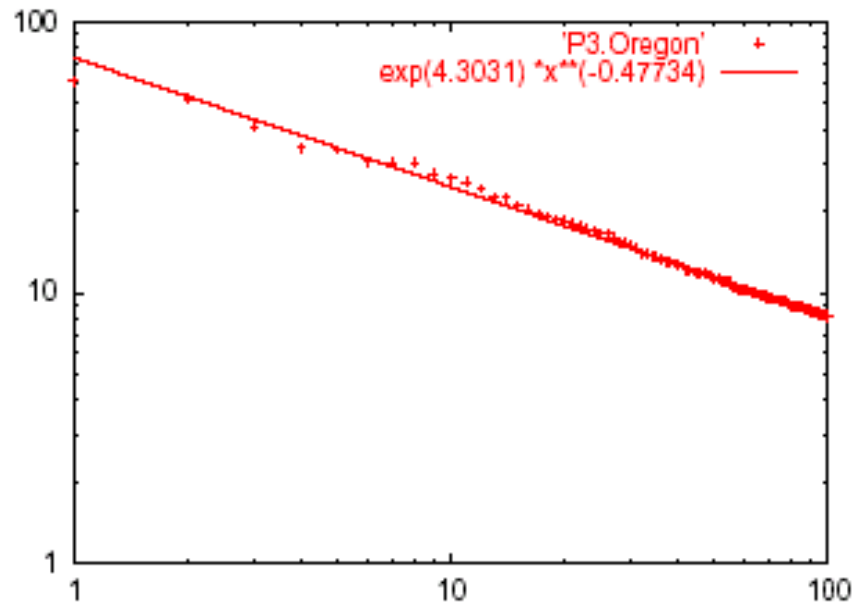
- Power law in the degree distribution [SIGCOMM99]

internet domains



# Solution# S.2: Eigen Exponent $E$

Eigenvalue



Exponent = slope

$$E = -0.48$$

May 2001

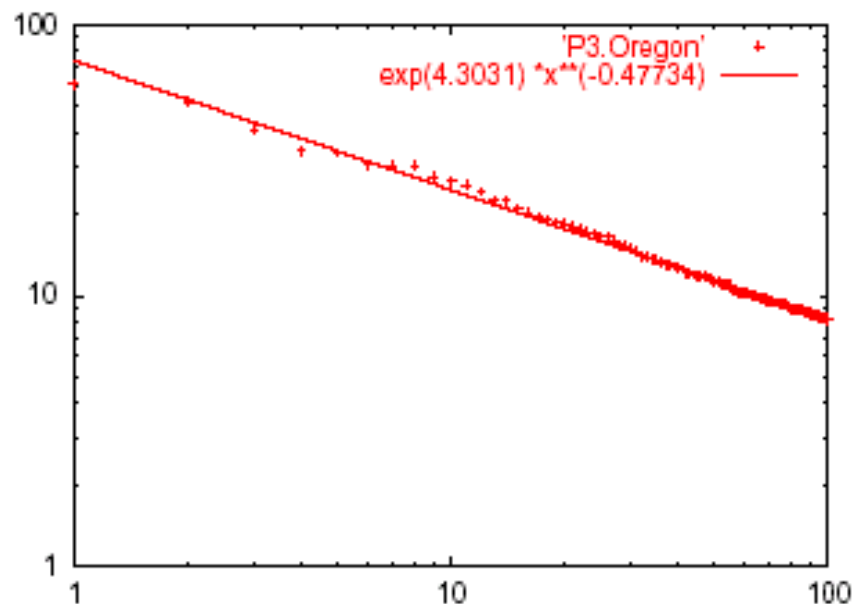
Rank of decreasing eigenvalue

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



# Solution# S.2: Eigen Exponent $E$

Eigenvalue



Exponent = slope

$$E = -0.48$$

May 2001

Rank of decreasing eigenvalue

- [Mihail, Papadimitriou '02]: slope is  $\frac{1}{2}$  of rank exponent

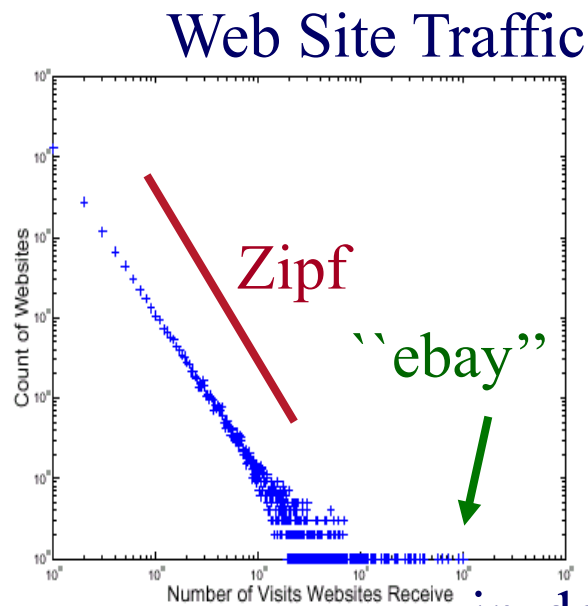
**But:**

How about graphs from other domains?

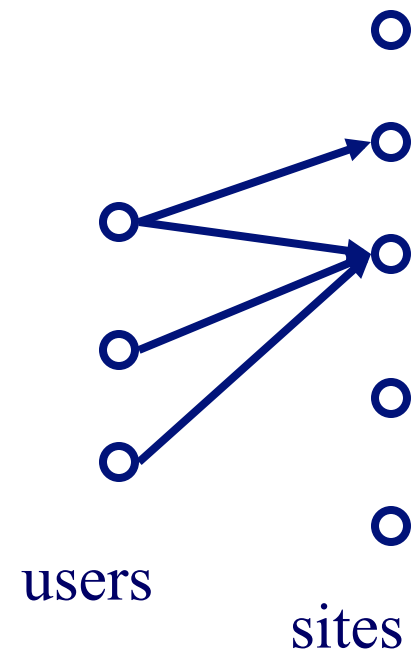
# More power laws:

- web hit counts [w/ A. Montgomery]

Count  
(log scale)

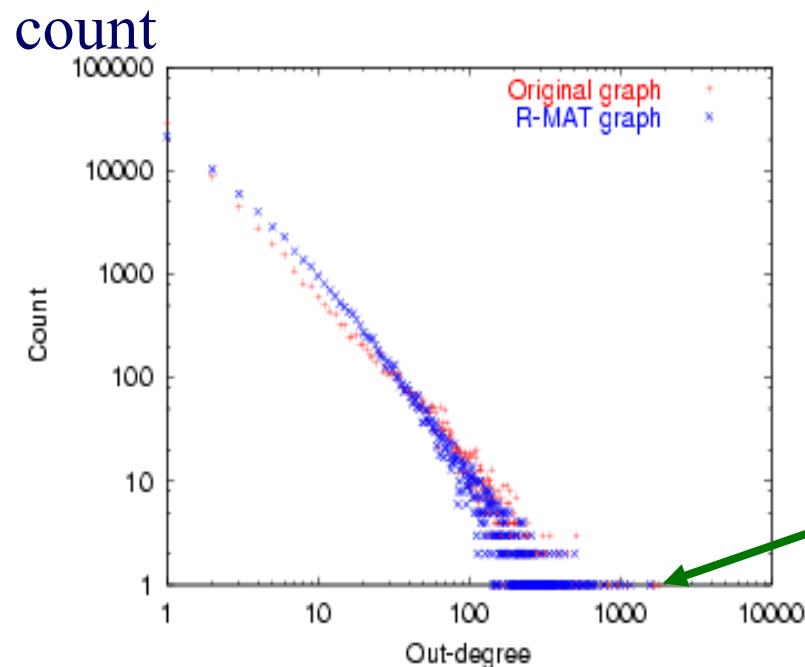


in-degree (log scale)



# epinions.com

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



trusts-2000-people user

(out) degree

## And numerous more

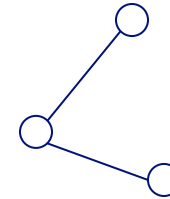
- # of sexual contacts
- Income [Pareto] – ‘80-20 distribution’
- Duration of downloads [Bestavros+]
- Duration of UNIX jobs (‘mice and elephants’)
- Size of files of a user
- ...
- ‘Black swans’

# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - triangles
    - cliques
  - Weighted graphs
  - Time evolving graphs
- Problem#2: Tools

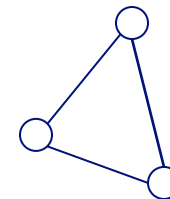


## Solution# S.3: Triangle ‘Laws’



- 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?

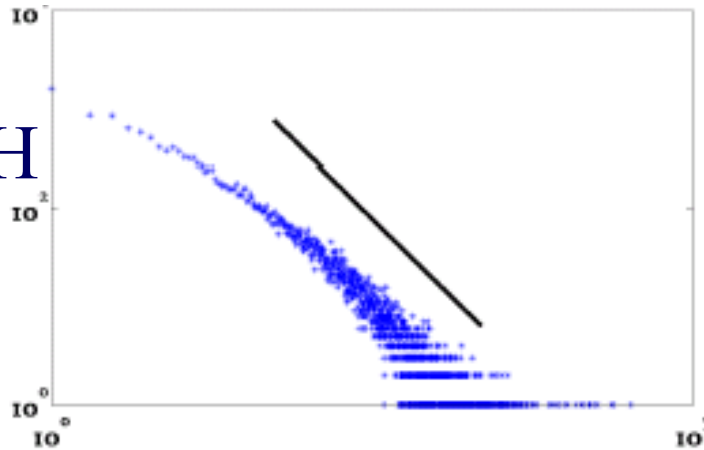


# Triangle Law: #S.3

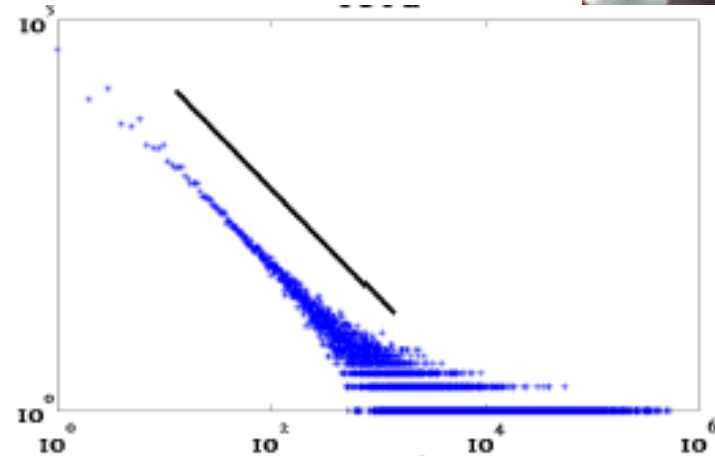
[Tsourakakis ICDM 2008]



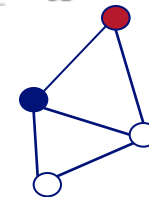
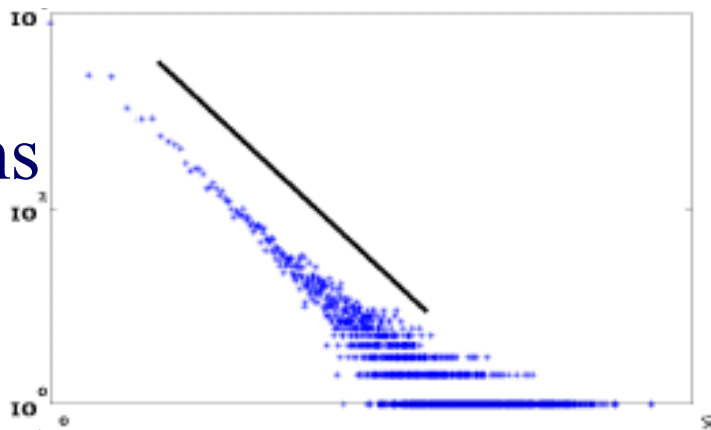
HEP-TH



ASN



Epinions



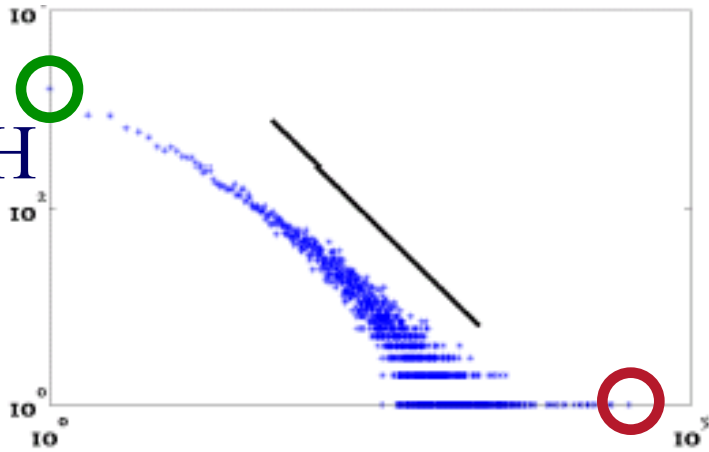
X-axis: # of participating triangles  
Y: count ( $\sim$  pdf)

# Triangle Law: #S.3

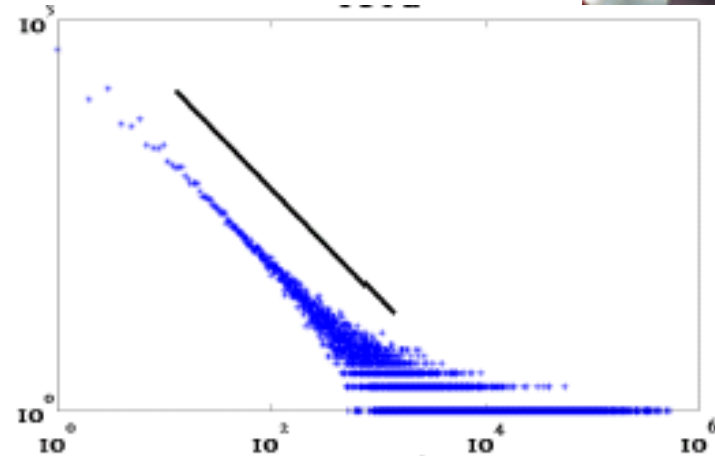
[Tsourakakis ICDM 2008]



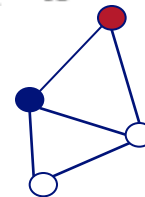
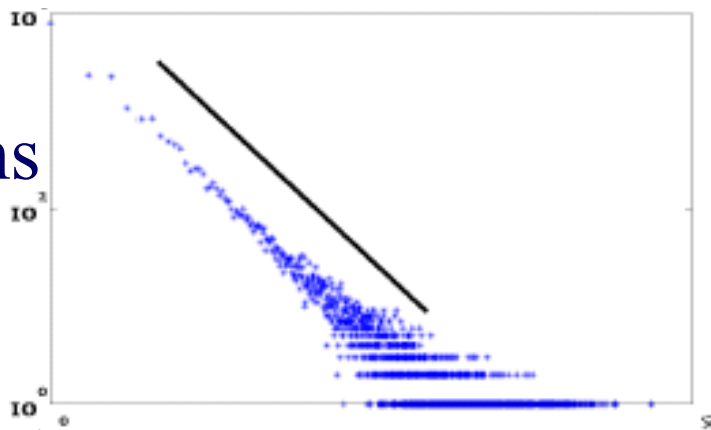
HEP-TH



ASN



Epinions

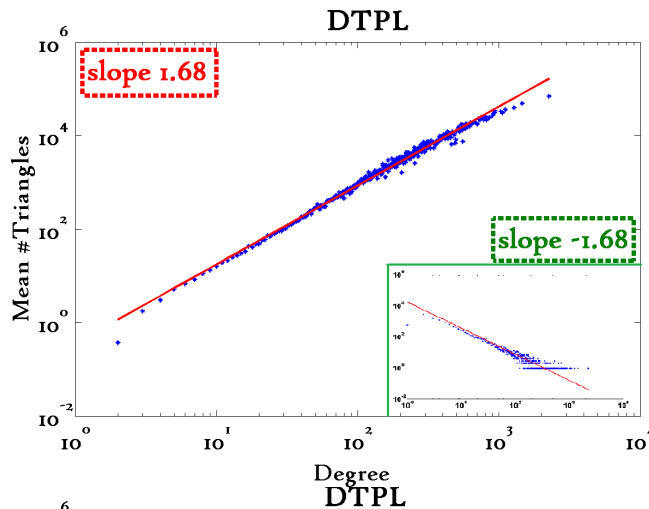


X-axis: # of participating triangles  
 Y: count ( $\sim$  pdf)

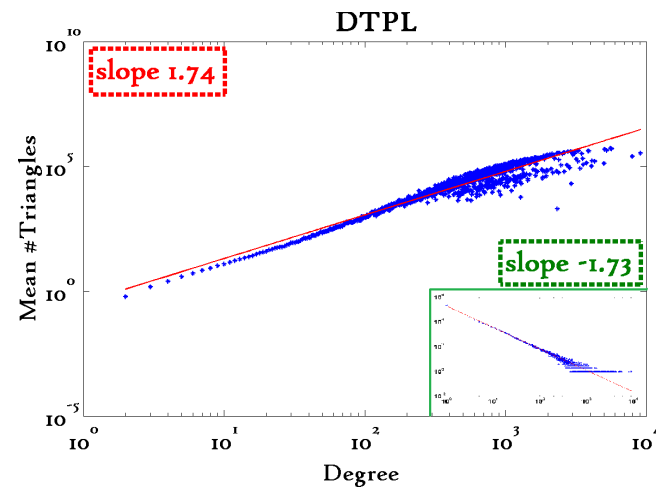
# Triangle Law: #S.4

## [Tsourakakis ICDM 2008]

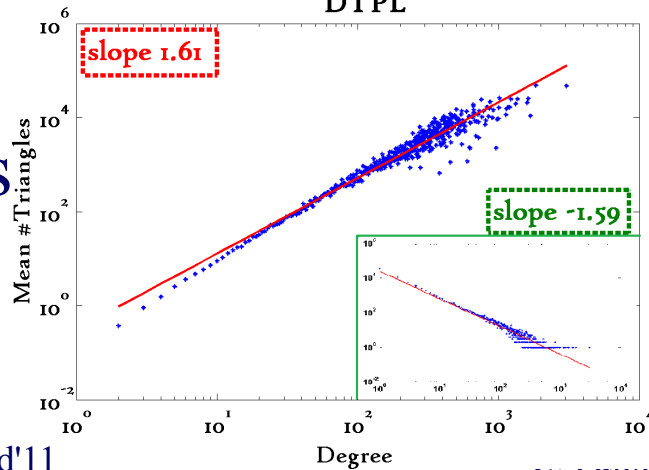
Reuters



SN



Epinions



X-axis: degree  
 Y-axis: mean # triangles  
 $n$  friends  $\rightarrow \sim n^{1.6}$  triangles

# Triangle Law: Computations

[Tsourakakis ICDM 2008]

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

# 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!

$$\#triangles = 1/6 \text{ Sum } ( \lambda_i^3 )$$

(and, because of skewness (S2) ,

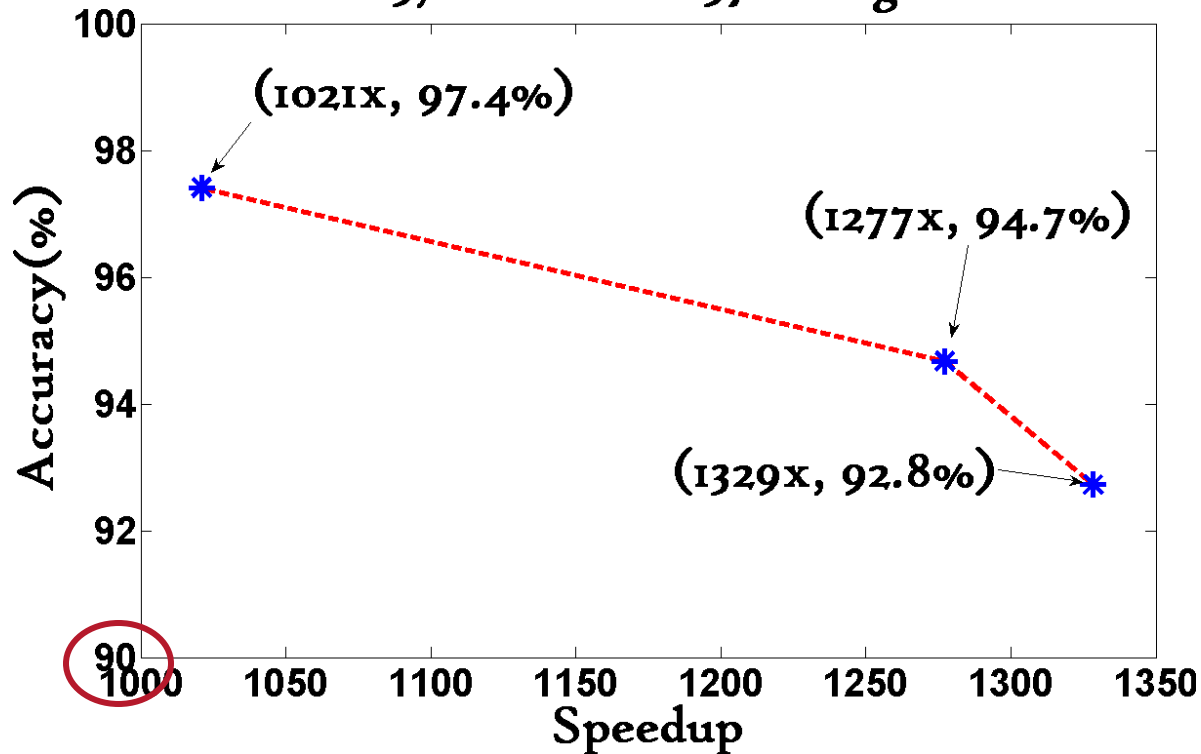
we only need the top few eigenvalues!

# Triangle Law: Computations

[Tsourakakis ICDM 2008]

Wikipedia graph 2006-Nov-04

$\approx 3,1\text{M}$  nodes  $\approx 37\text{M}$  edges



1000x+ speed-up, >90% accuracy

# EigenSpokes

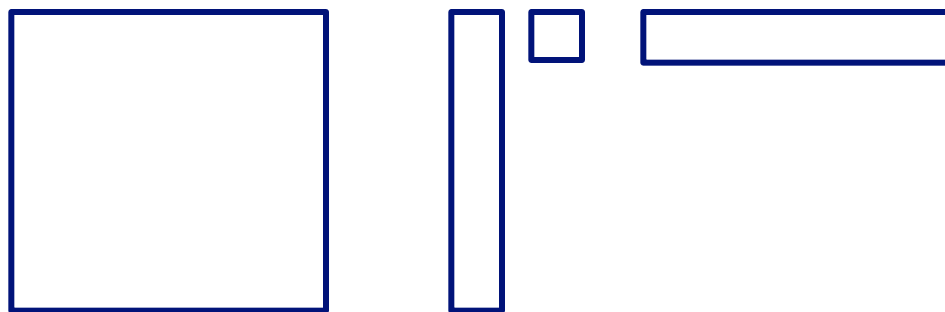


B. Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju and Christos Faloutsos: *EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs*, PAKDD 2010, Hyderabad, India, 21-24 June 2010.

# EigenSpokes

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

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





# EigenSpokes

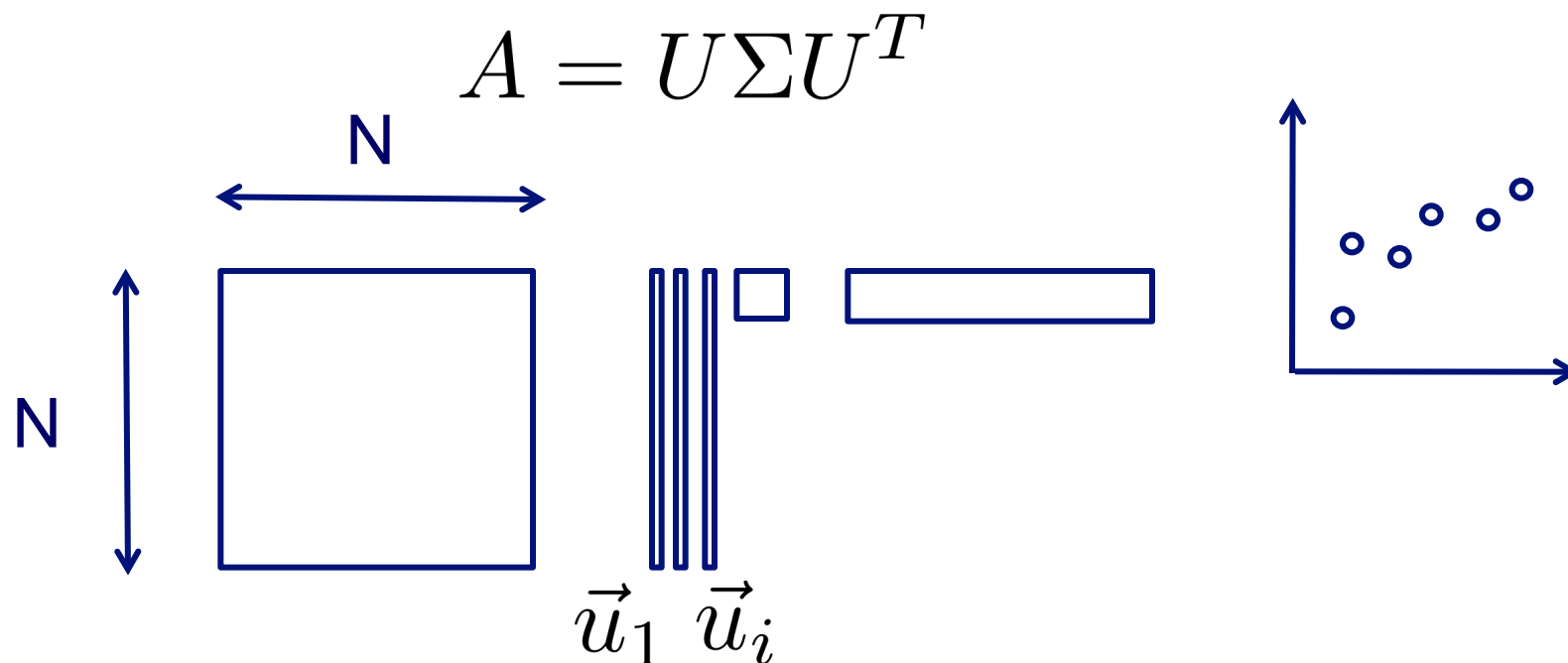
- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

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

The diagram illustrates the SVD decomposition of an adjacency matrix  $A$ . On the left, a square matrix  $A$  is shown with a vertical double-headed arrow labeled  $N$  and a horizontal double-headed arrow labeled  $N$ . To its right, the equation  $A = U \Sigma U^T$  is displayed. Matrix  $U$  is represented by a tall, narrow rectangle with three vertical lines, labeled with vectors  $\vec{u}_1$  and  $\vec{u}_i$  below it. Matrix  $\Sigma$  is a small square. Matrix  $U^T$  is a wide, short rectangle.

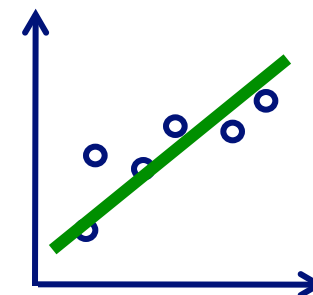
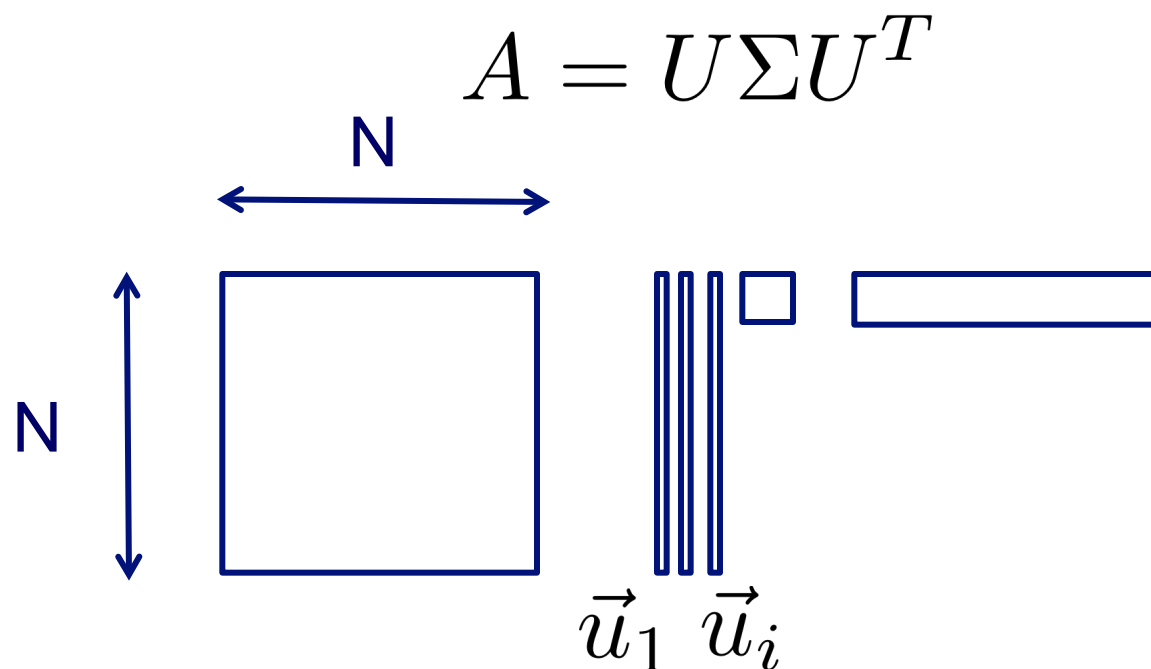
# EigenSpokes

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



# EigenSpokes

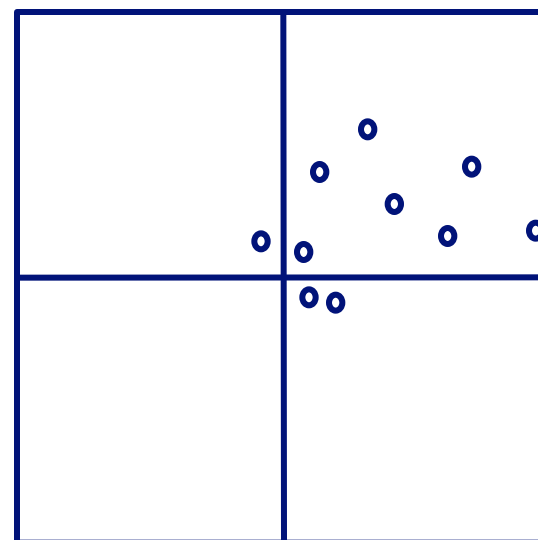
- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)



# EigenSpokes

- EE plot:
- Scatter plot of scores of  $u_1$  vs  $u_2$
- One would expect
  - Many points @ origin
  - A few scattered ~randomly

2<sup>nd</sup> Principal component  
 $u_2$

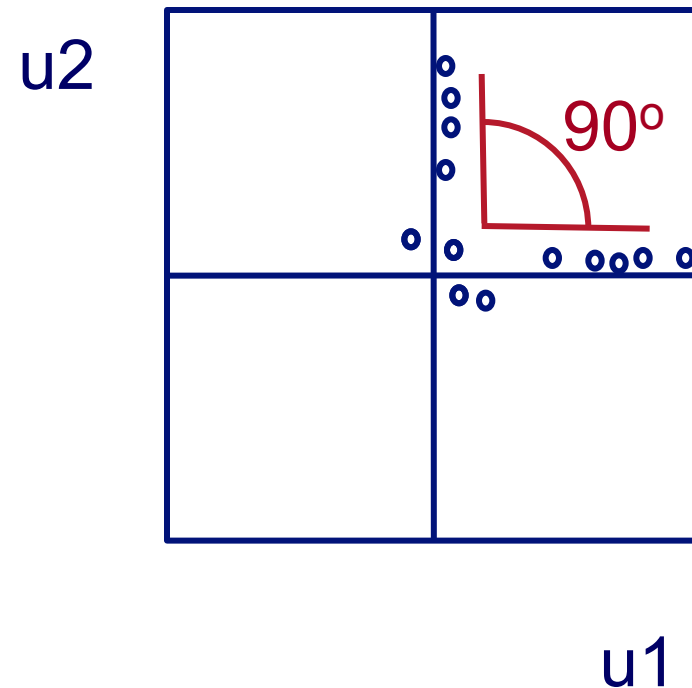


$u_1$

1<sup>st</sup> Principal component

# EigenSpokes

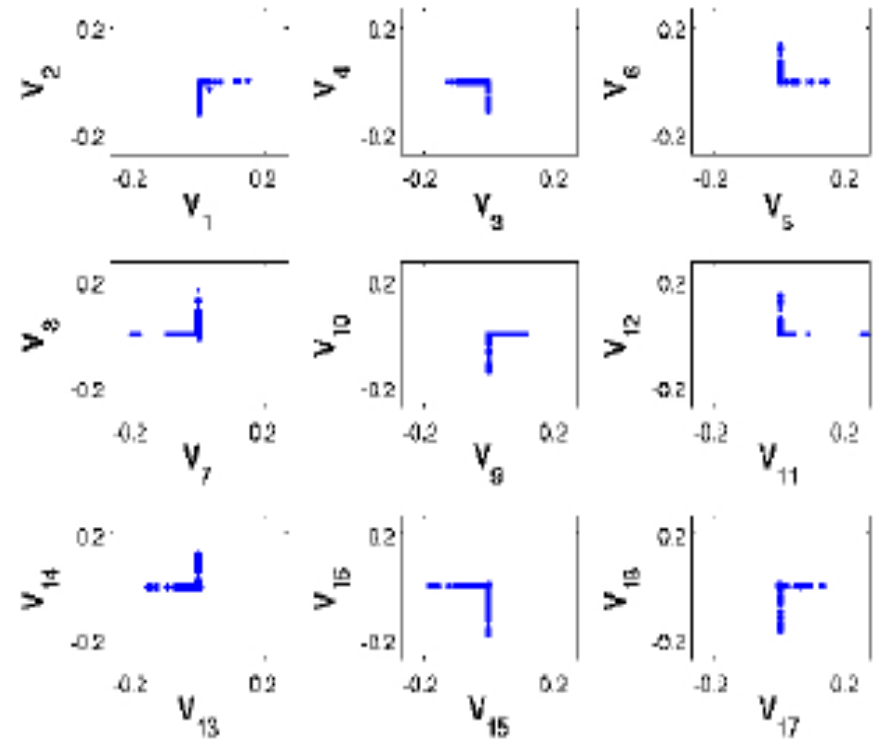
- EE plot:
- Scatter plot of scores of  $u_1$  vs  $u_2$
- One would expect
  - Many points @ origin
  - A few scattered  $\sim$  random



# EigenSpokes - pervasiveness

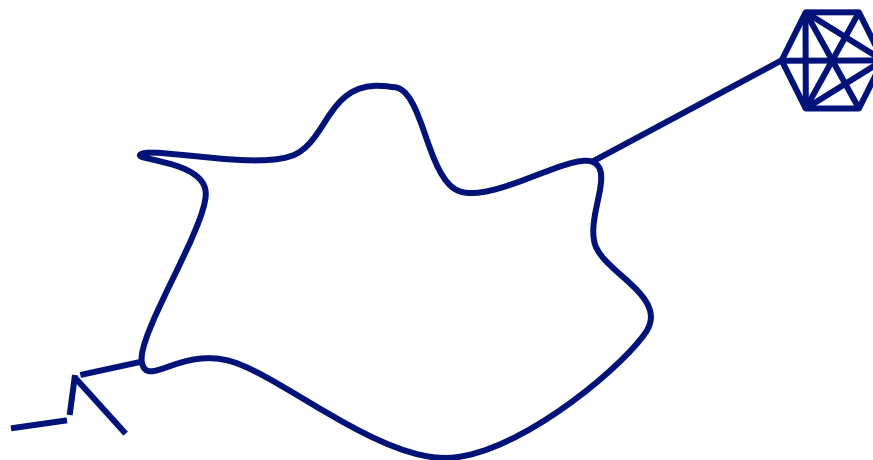
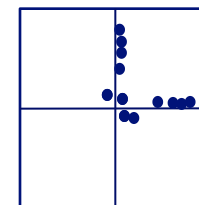
- Present in mobile social graph
  - across time and space

- Patent citation graph



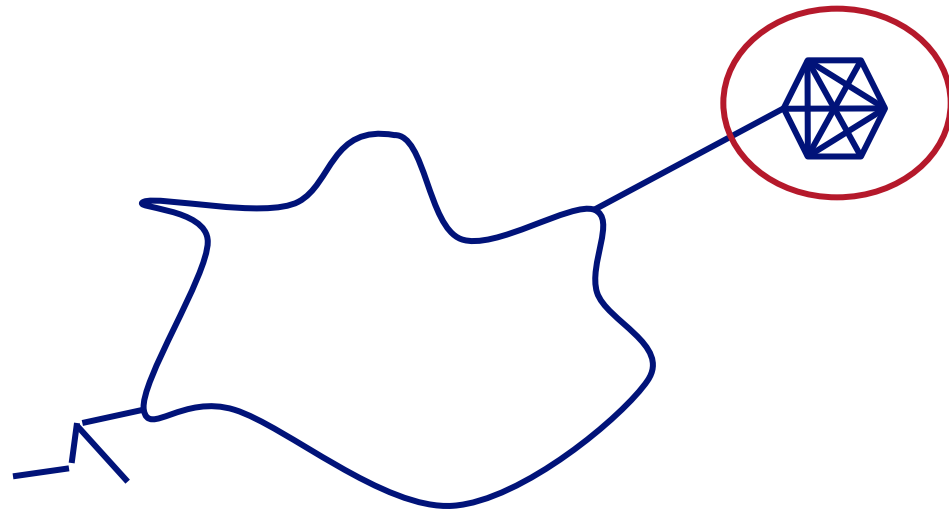
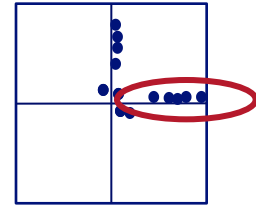
# EigenSpokes - explanation

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



# EigenSpokes - explanation

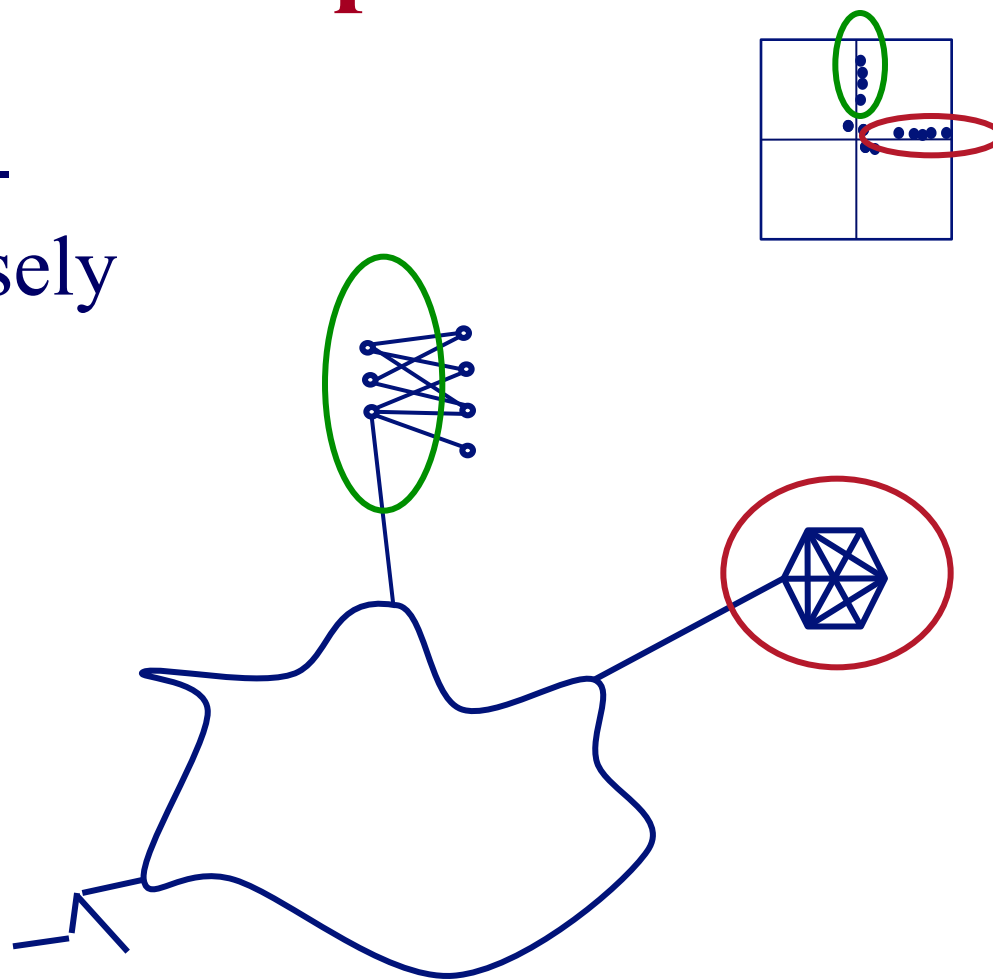
Near-cliques, or near-bipartite-cores, loosely connected





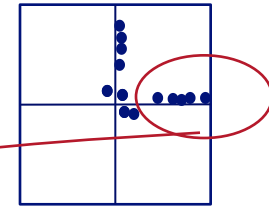
# EigenSpokes - explanation

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

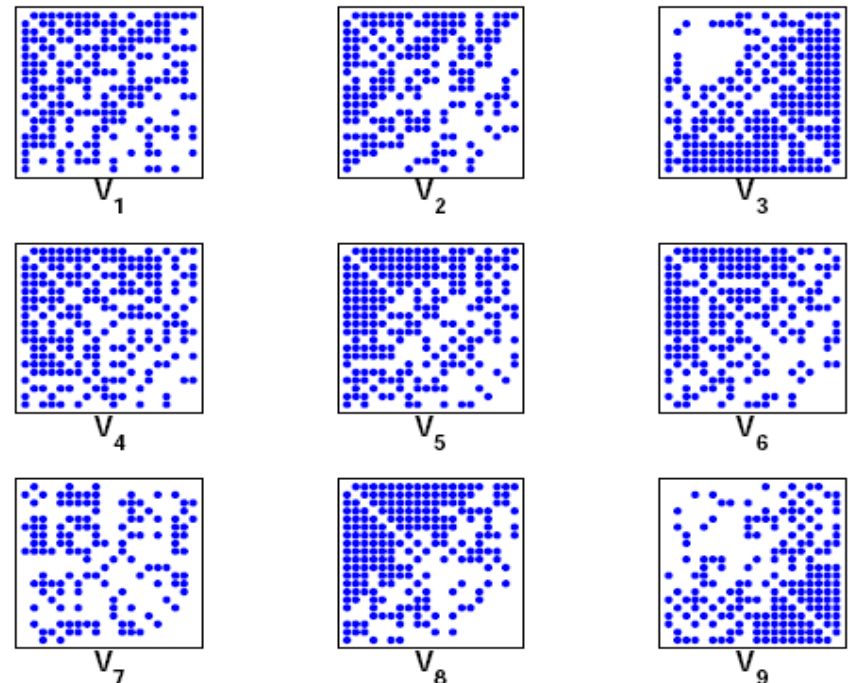


# EigenSpokes - explanation

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



spy plot of top 20 nodes



So what?

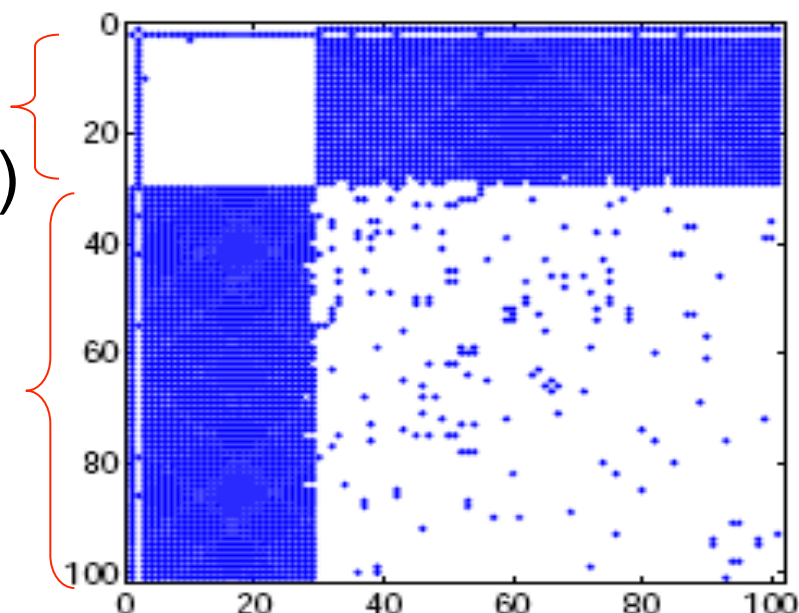
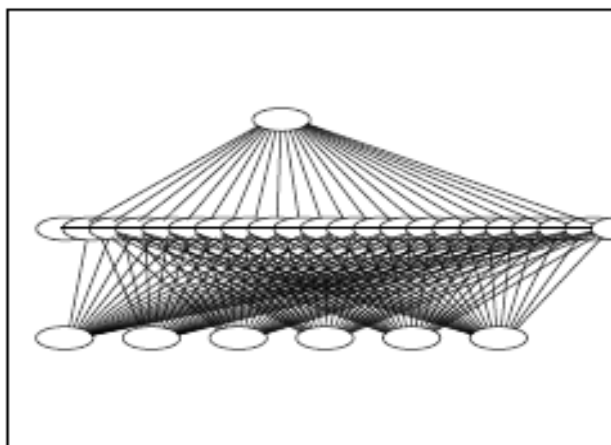
- Extract nodes with high *scores*
- high connectivity
- Good “communities”

# Bipartite Communities!

patents from  
same inventor(s)

`cut-and-paste'  
bibliography!

magnified bipartite community



# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - triangles
    - cliques
  - ➔ – Weighted graphs
  - Time evolving graphs
- Problem#2: Tools

# Observations on weighted graphs?

- A: yes - even more 'laws'!



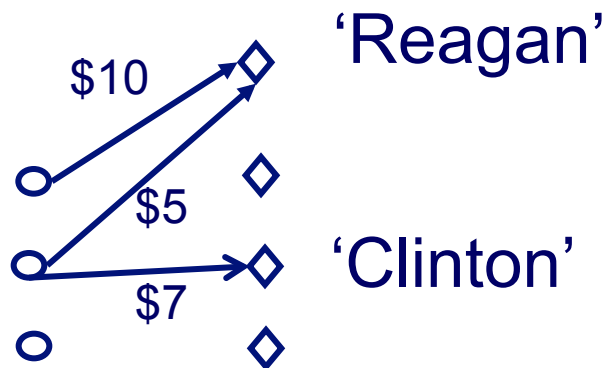
M. McGlohon, L. Akoglu, and C. Faloutsos  
*Weighted Graphs and Disconnected  
Components: Patterns and a Generator.*  
*SIG-KDD 2008*

## Observation W.1: Fortification

*Q: How do the weights  
of nodes relate to degree?*

# Observation W.1: Fortification

**More donors,  
more \$ ?**



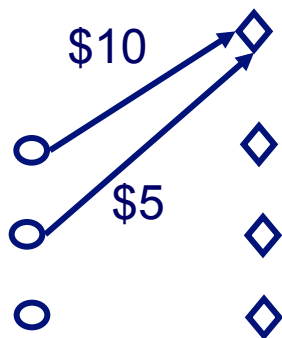
Stanford'11

C. Faloutsos (CMU)

# Observation W.1: fortification: Snapshot Power Law

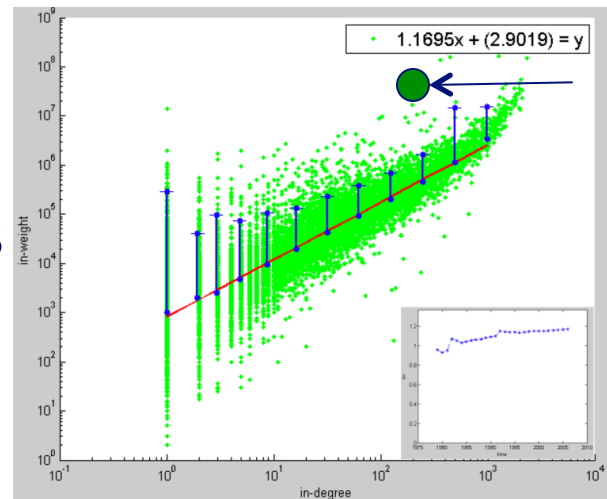
- Weight: super-linear on in-degree
- exponent 'iw':  $1.01 < iw < 1.26$

**More donors,  
even more \$**



Stanford'11

In-weights  
(\$)



Edges (# donors)

**Orgs-Candidates**

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

C. Faloutsos (CMU)



# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
  - Weighted graphs
  - ➔ – Time evolving graphs
- Problem#2: Tools
- ...

# Problem: Time evolution

- with Jure Leskovec (CMU -> Stanford)
- and Jon Kleinberg (Cornell – sabb. @ CMU)

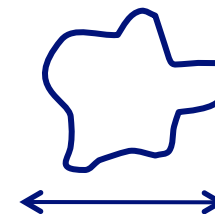
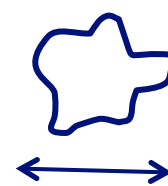


## T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:

- diameter  $\sim O(\log N)$

- diameter  $\sim O(\log \log N)$



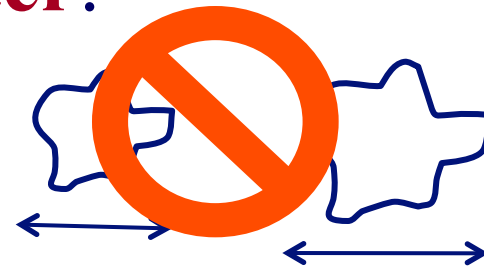
- What is happening in real data?

# T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:

- diameter  $\sim O(\log N)$

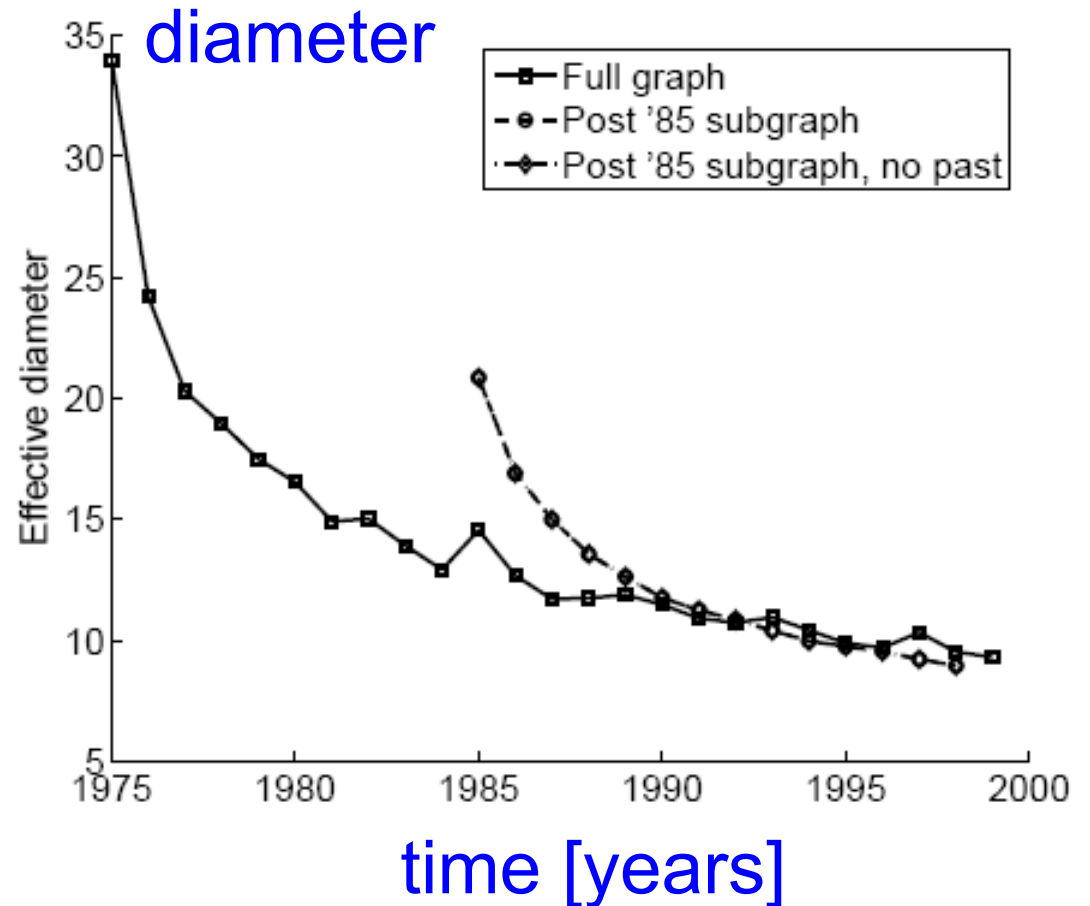
- diameter  $\sim O(\log \log N)$



- 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

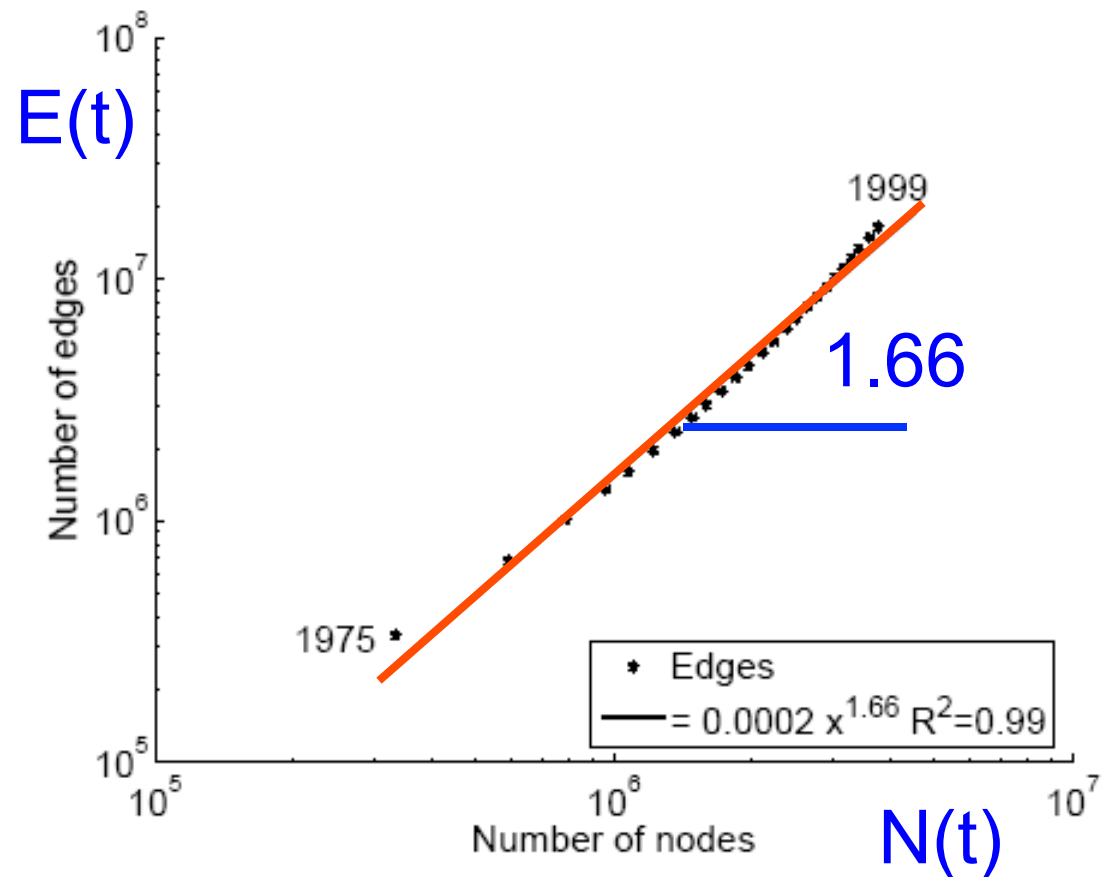
$$E(t+1) = \text{?} * E(t)$$

- 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





# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
  - Weighted graphs
  - ➔ – Time evolving graphs
- Problem#2: Tools
- ...

# More on Time-evolving graphs

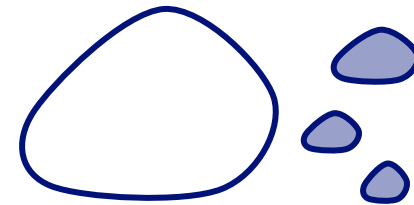
M. McGlohon, L. Akoglu, and C. Faloutsos  
*Weighted Graphs and Disconnected  
Components: Patterns and a Generator.*  
*SIG-KDD 2008*

## 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?

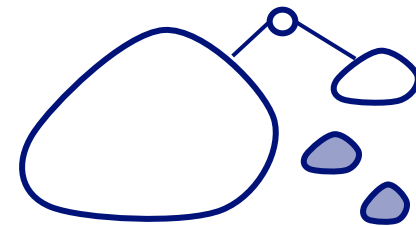


## Observation T.3: NLCC behavior

*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 shrink?
- or stabilize?



## Observation T.3: NLCC behavior

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

(“NLCC” = non-largest conn. components)

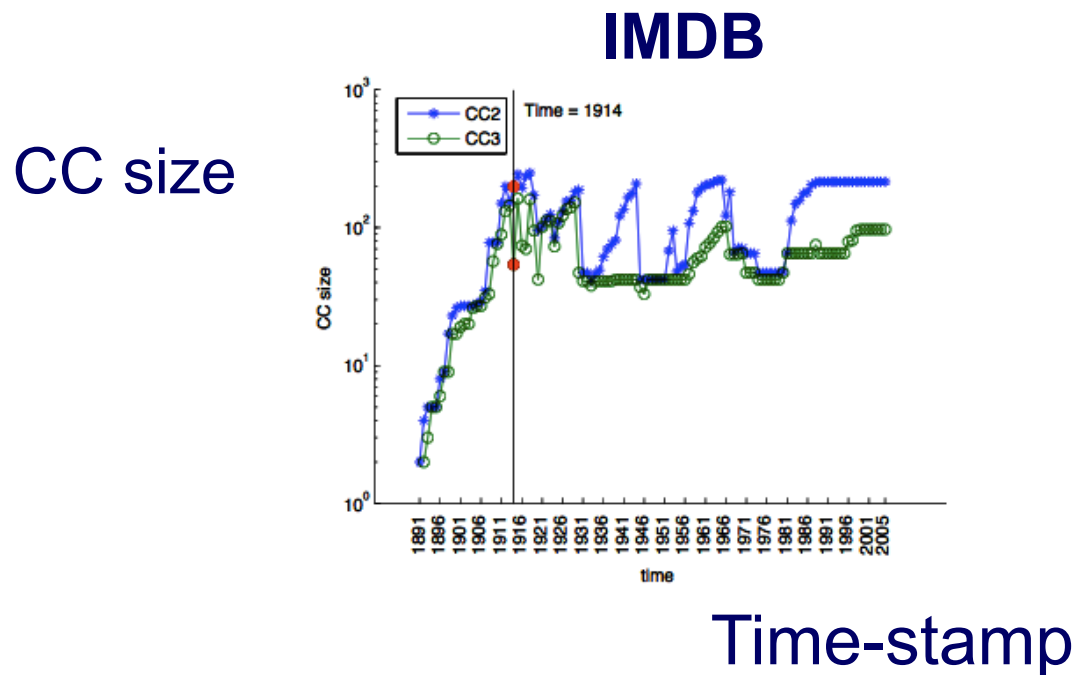
**YES** – Do they continue to grow in size?

**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  $\sim$ constant (actually, **oscillate**).

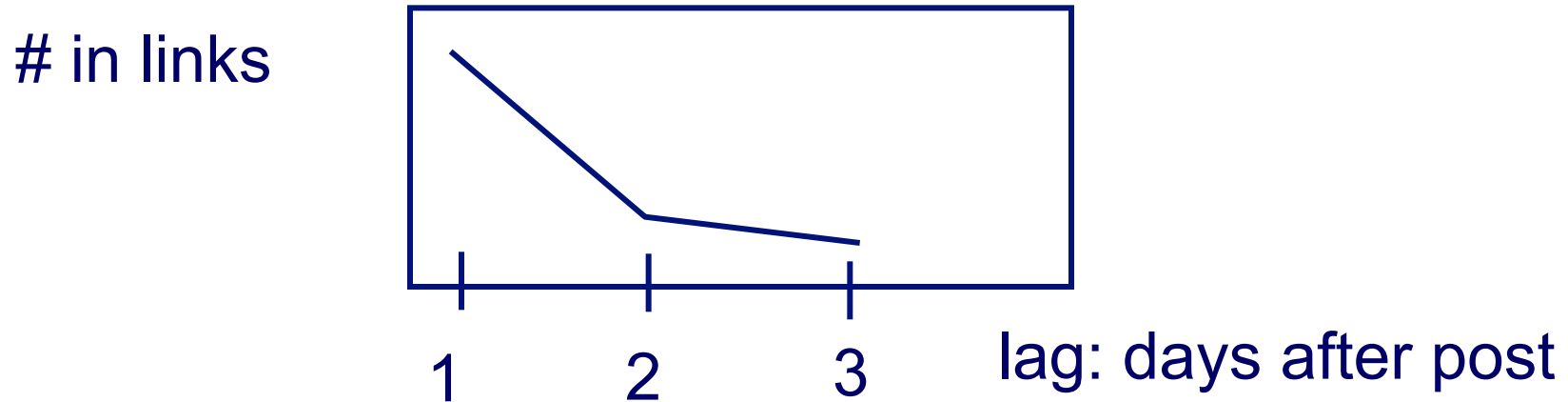


## Timing for Blogs

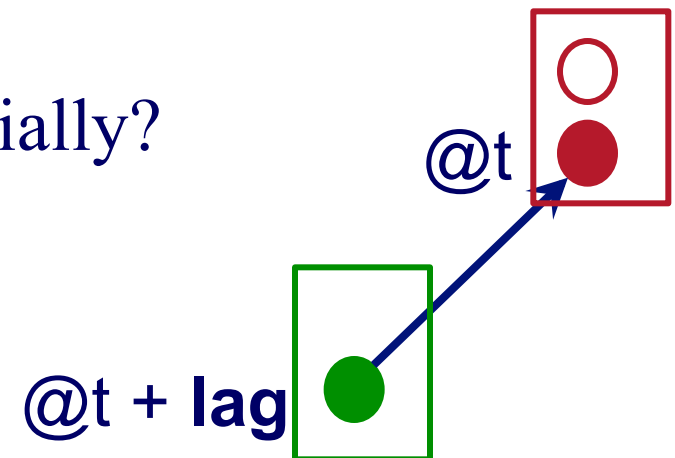
- with Mary McGlohon (CMU->google)
- Jure Leskovec (CMU->Stanford)
- Natalie Glance (now at Google)
- Mat Hurst (now at MSR)

[SDM'07]

## T.4 : popularity over time



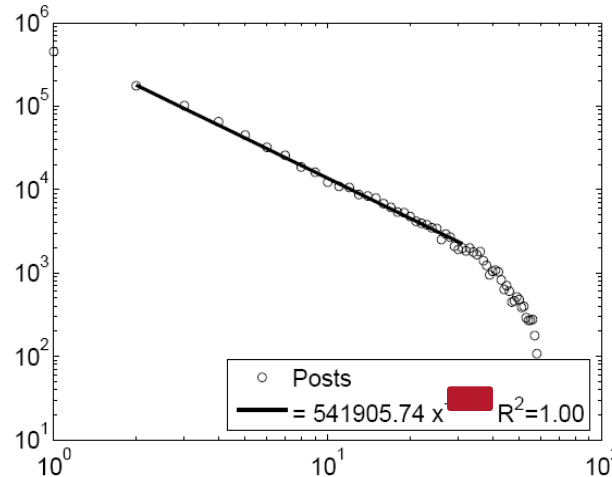
Post popularity drops-off – exponentially?





# T.4 : popularity over time

# in links  
(log)

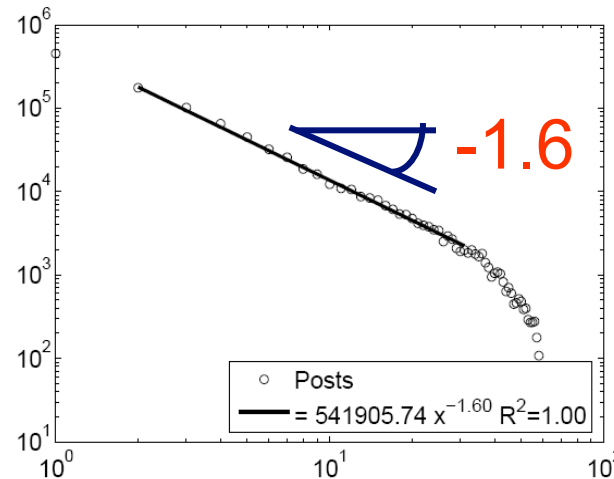


days after post  
(log)

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?

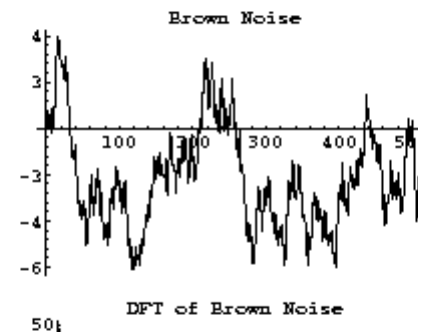
POWER LAW!

Exponent? -1.6

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

Stanford'11

C. Faloutsos (CMU)



## -1.5 slope

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

## T.5: duration of phonecalls

*Surprising Patterns for the Call  
Duration Distribution of Mobile  
Phone Users*

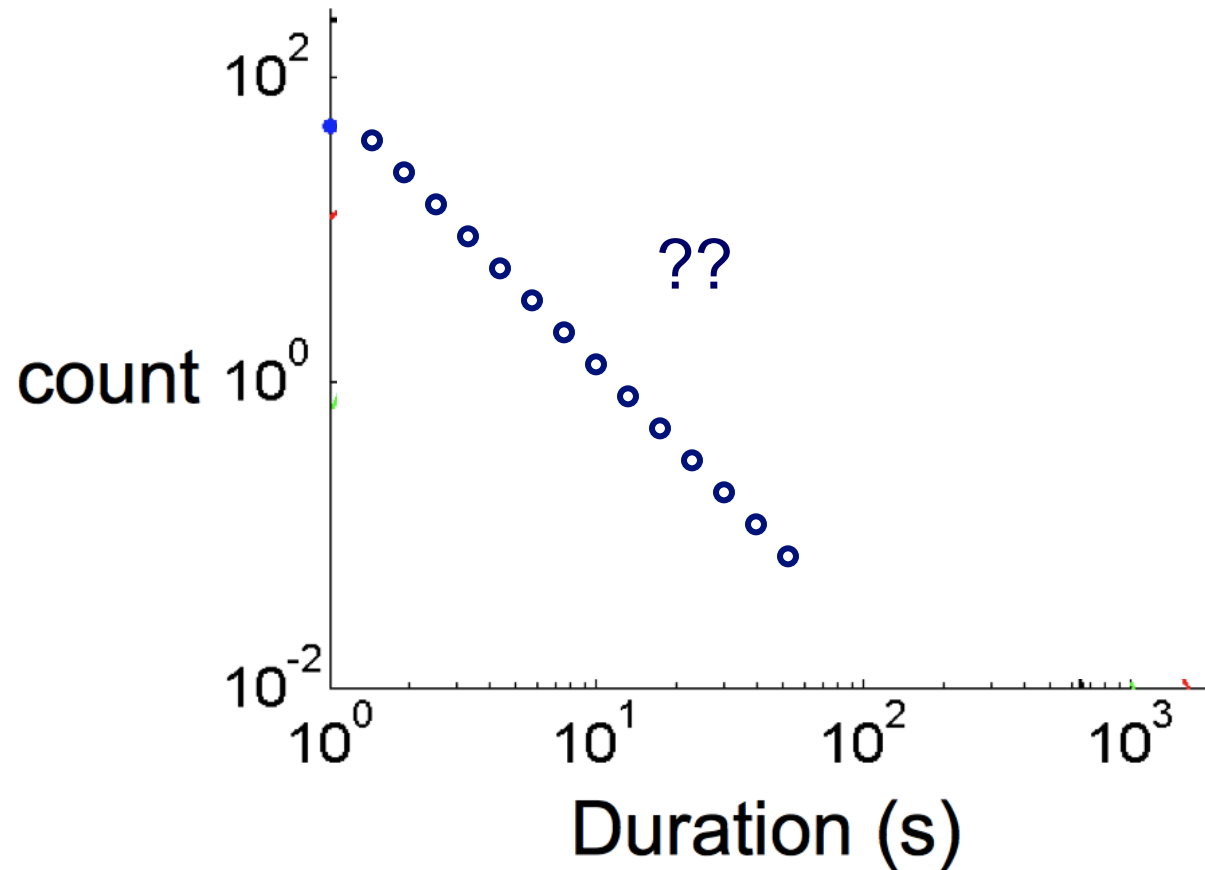


Pedro O. S. Vaz de Melo, Leman

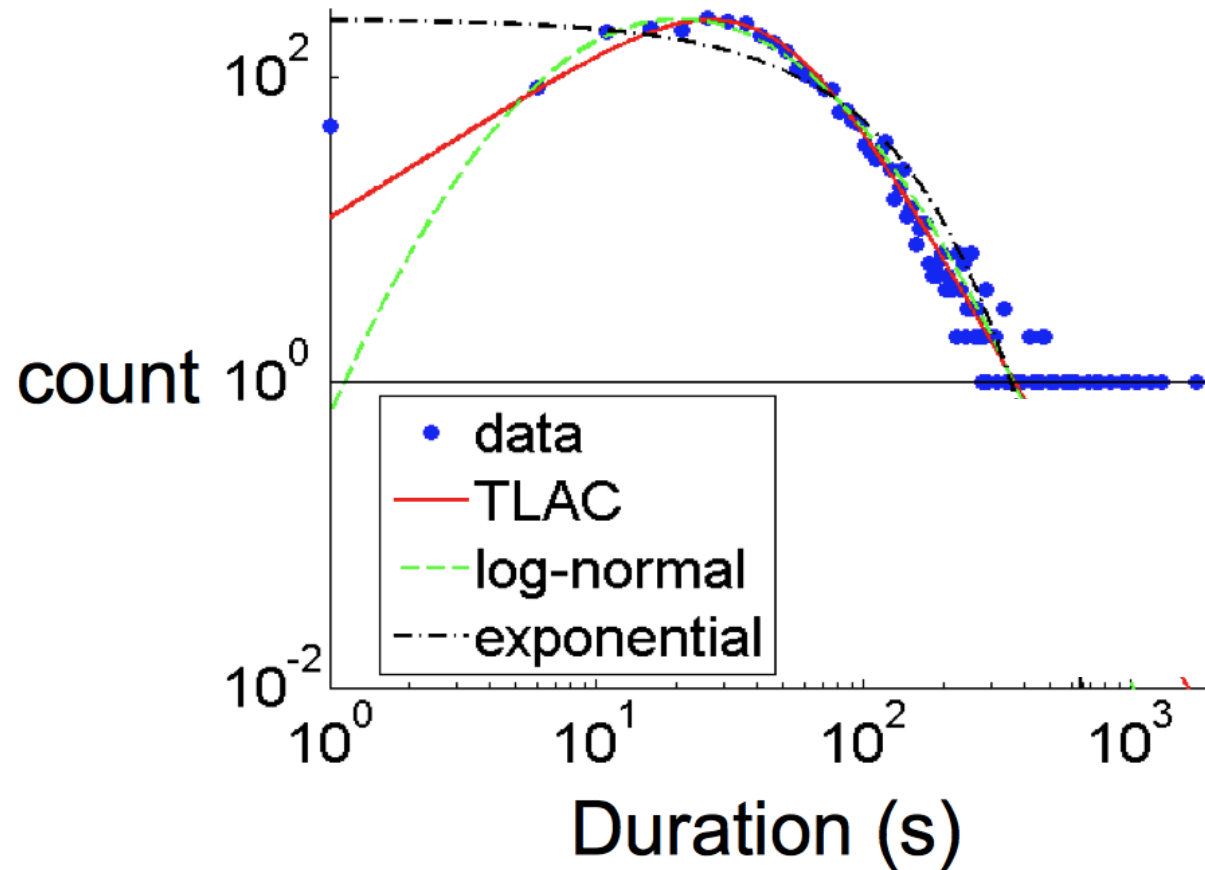
Akoglu, Christos Faloutsos, Antonio  
A. F. Loureiro

PKDD 2010

# Probably, power law (?)

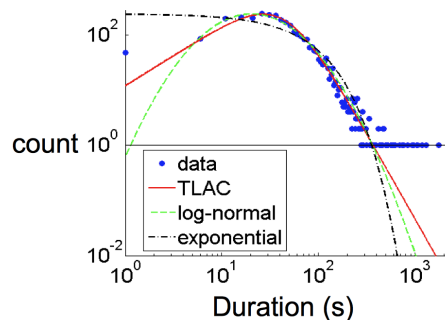


# No Power Law!



# 'TLaC: Lazy Contractor'

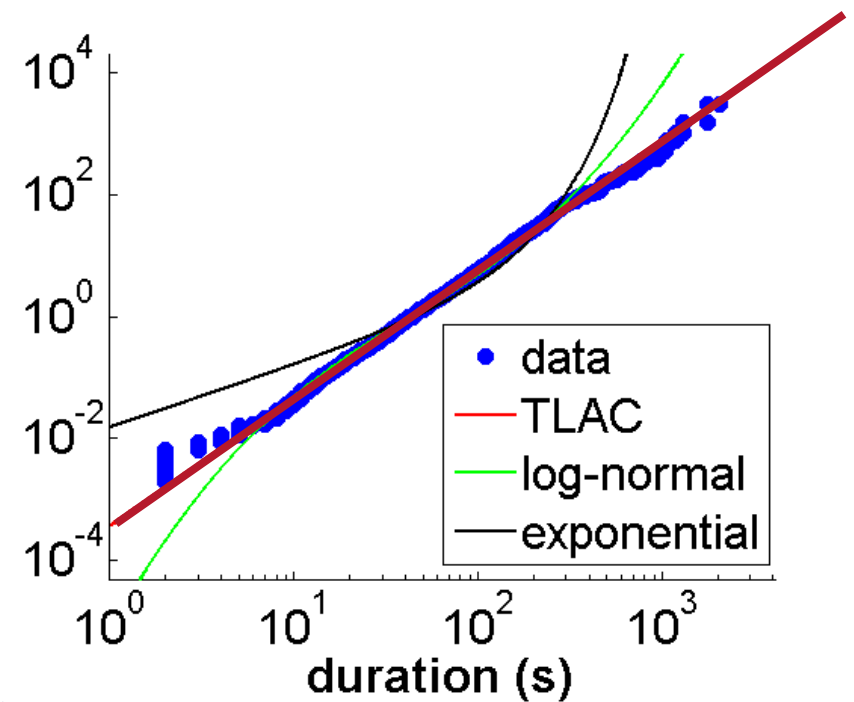
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

*Casualties*( $<x$ ):  
*Survivors*( $\geq x$ )

== 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)



# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - ➔ – OddBall (anomaly detection)
  - Belief Propagation
  - Immunization
- Problem#3: Scalability
- Conclusions

# OddBall: Spotting Anomalies in Weighted Graphs



Leman Akoglu, Mary McGlohon, Christos  
Faloutsos

*Carnegie Mellon University  
School of Computer Science*

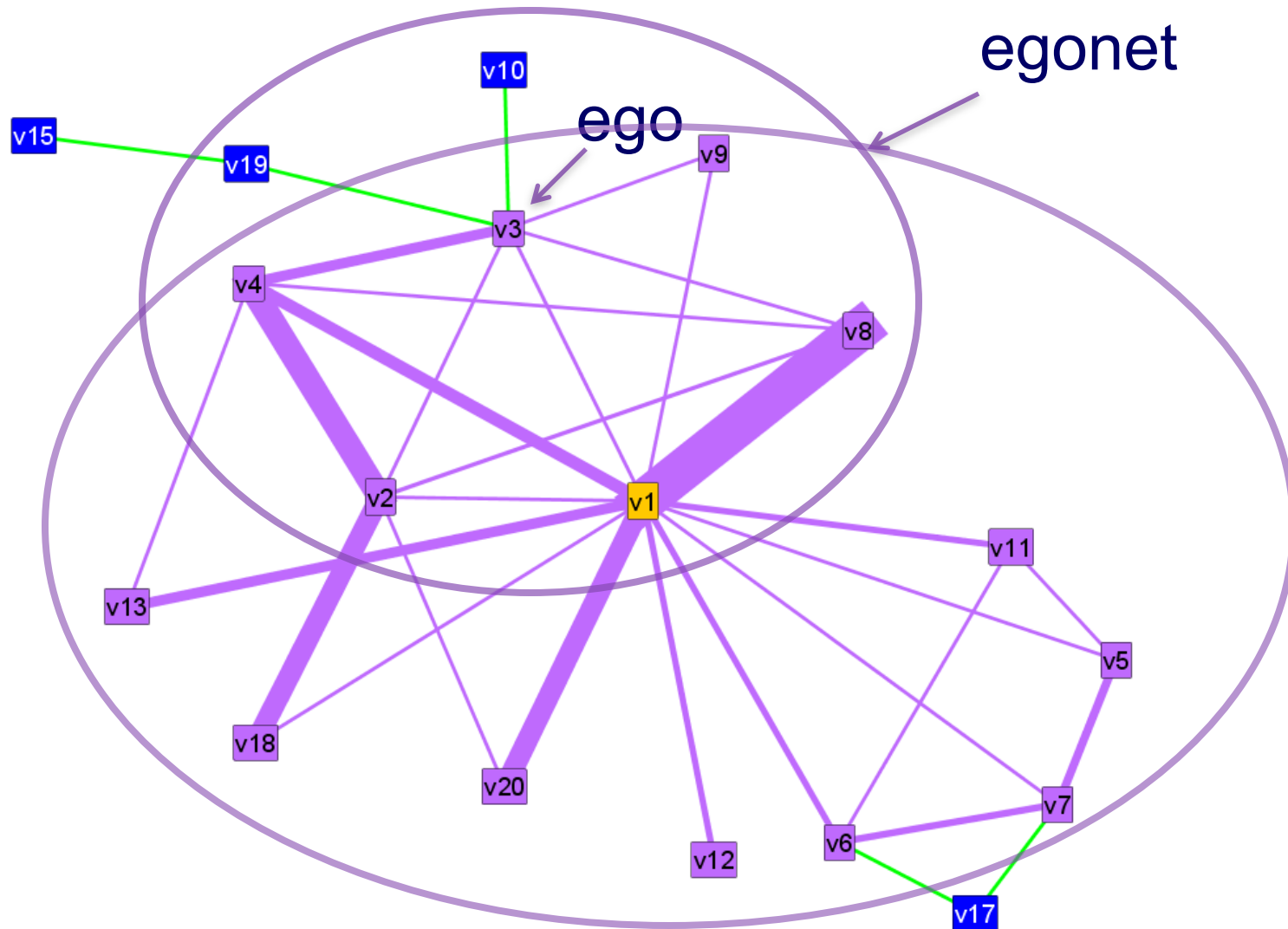
PAKDD 2010, Hyderabad, India

## Main idea

For each node,

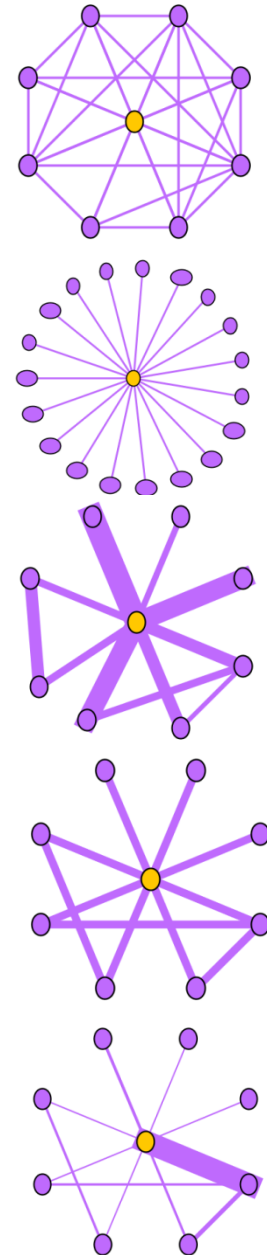
- extract ‘ego-net’ (=1-step-away neighbors)
- Extract features (#edges, total weight, etc etc)
- Compare with the rest of the population

# What is an egonet?

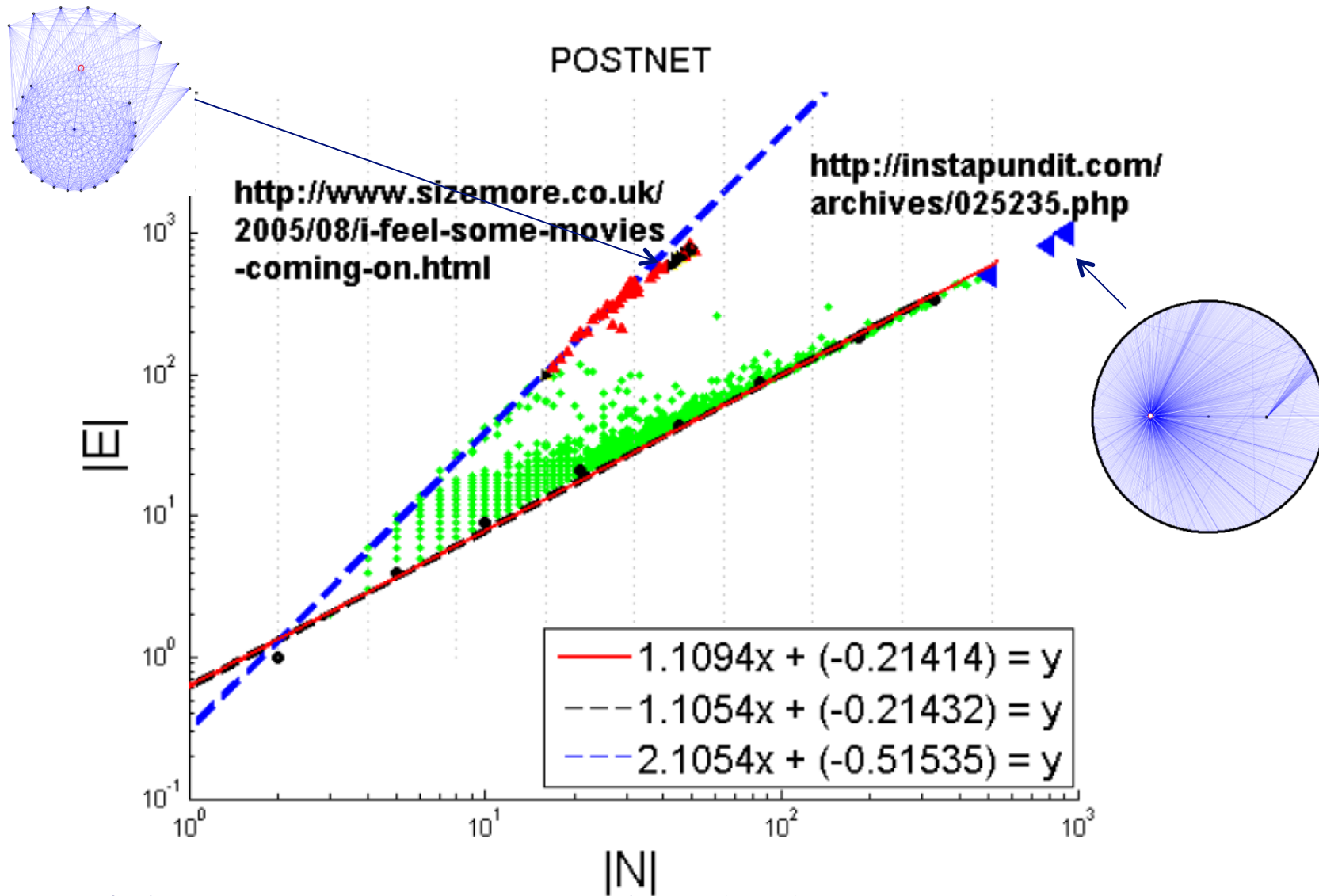


# Selected Features

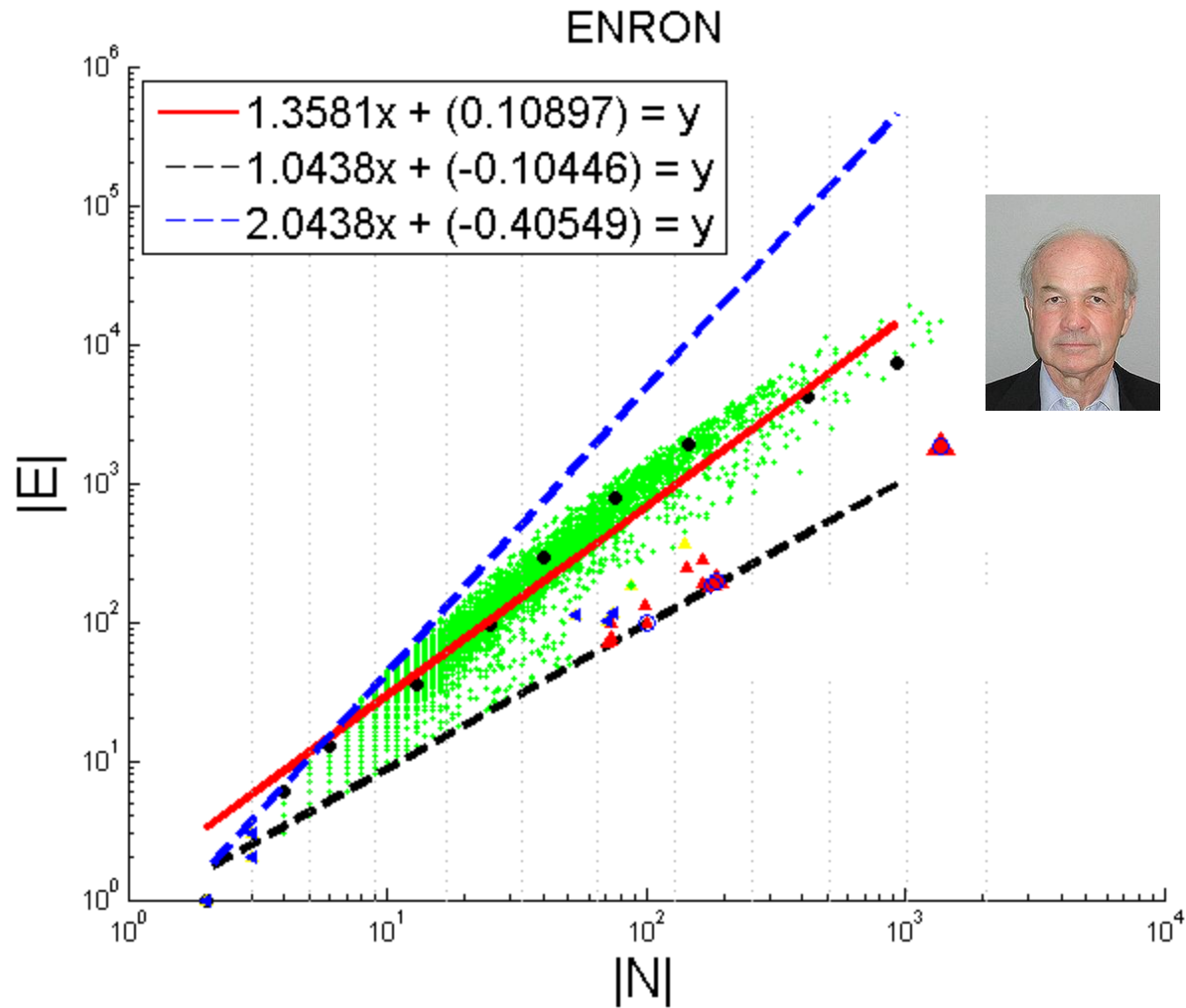
- $N_i$ : number of neighbors (degree) of ego  $i$
- $E_i$ : number of edges in egonet  $i$
- $W_i$ : total weight of egonet  $i$
- $\lambda_{w,i}$ : principal eigenvalue of the **weighted** adjacency matrix of egonet  $I$



# Near-Clique/Star



# Near-Clique/Star



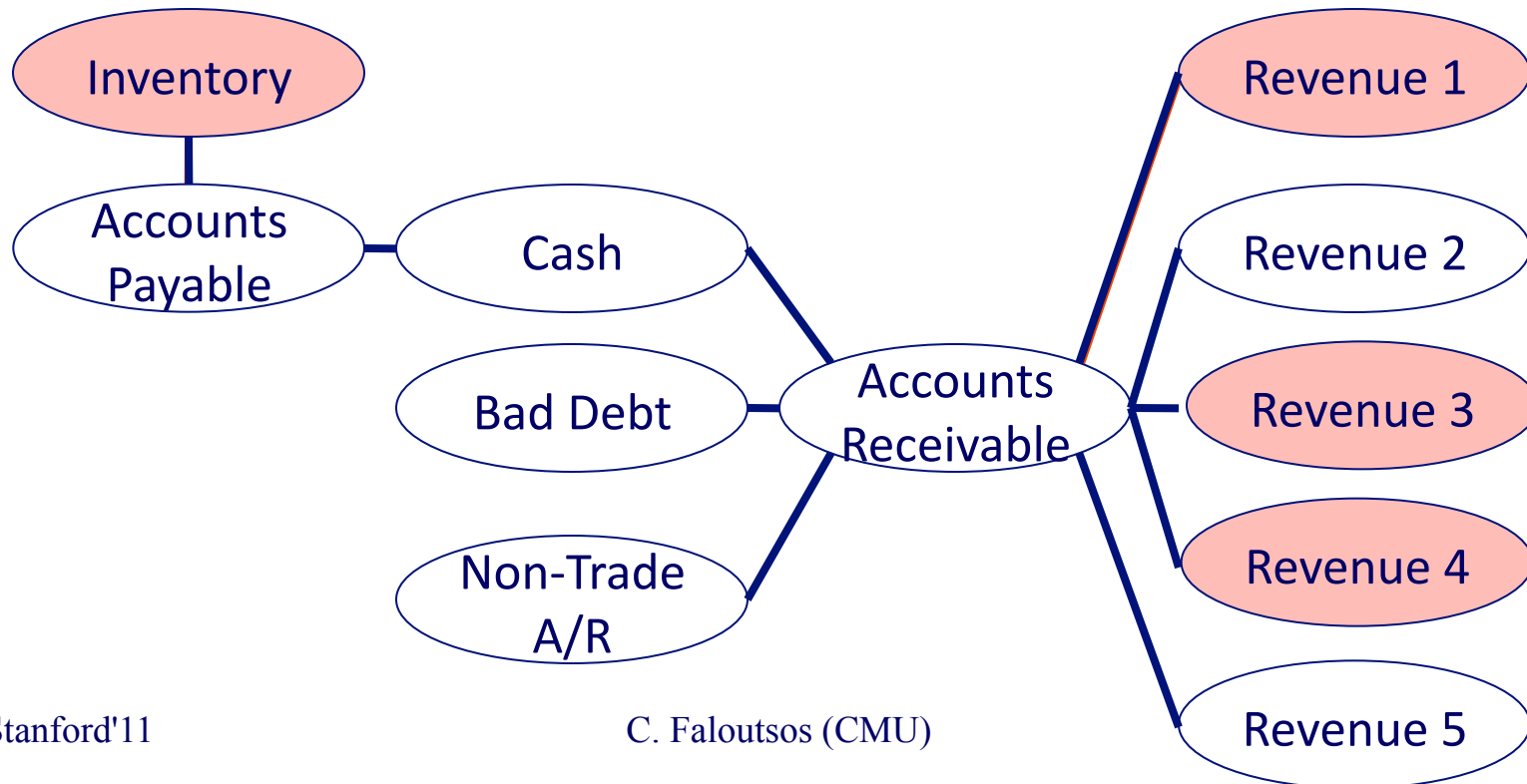
# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - OddBall (anomaly detection)
  - ➔ – Belief Propagation
  - Immunization
- Problem#3: Scalability
- Conclusions



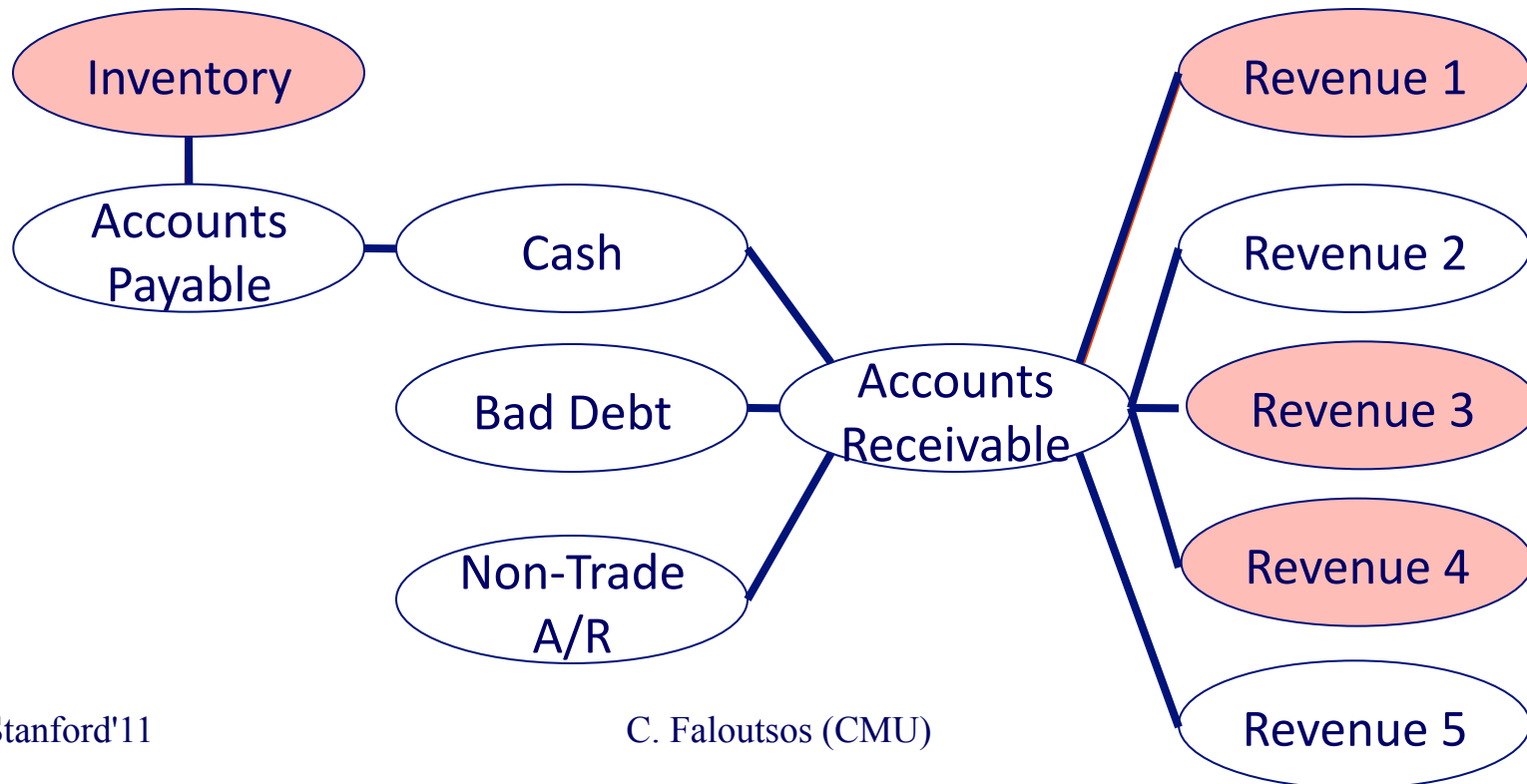
# Fraud detection

- **Problem:** Given network and noisy domain knowledge about weakly-suspicious nodes (flags), which nodes are most risky?



# Fraud detection

- Flags: eg, too many round numbers, etc



## Solution: Belief Propagation

- Solution: **S**ocial **N**etwork **A**nalytic **R**isk **E**valuation
  - Assume **homophily** between nodes (“guilt by association”)
  - Use **belief propagation** (message passing)
  - Upon convergence, determine **end risk scores**.

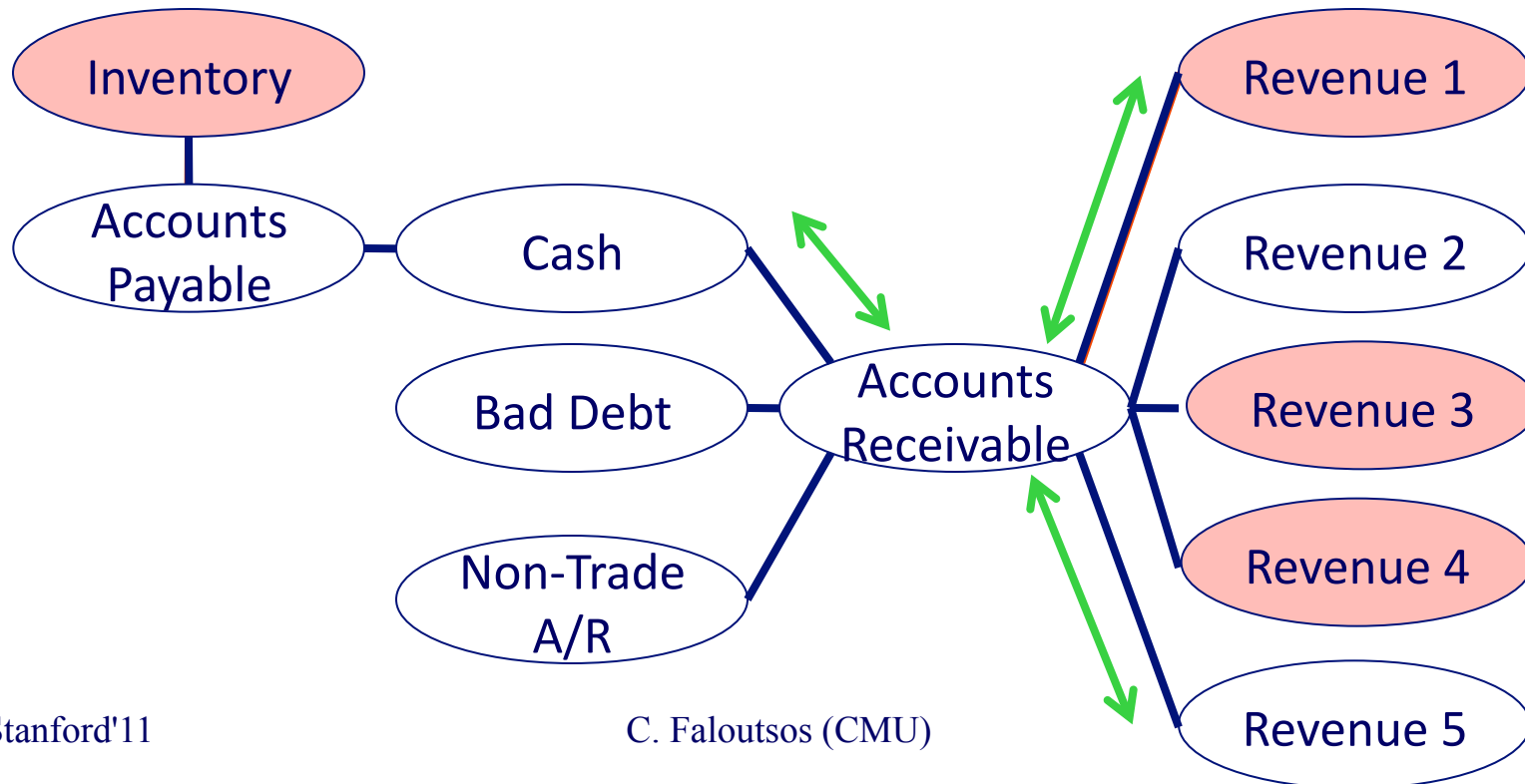
[SNARE: McGlohon+, KDD'09]

Stanford'11

C. Faloutsos (CMU)

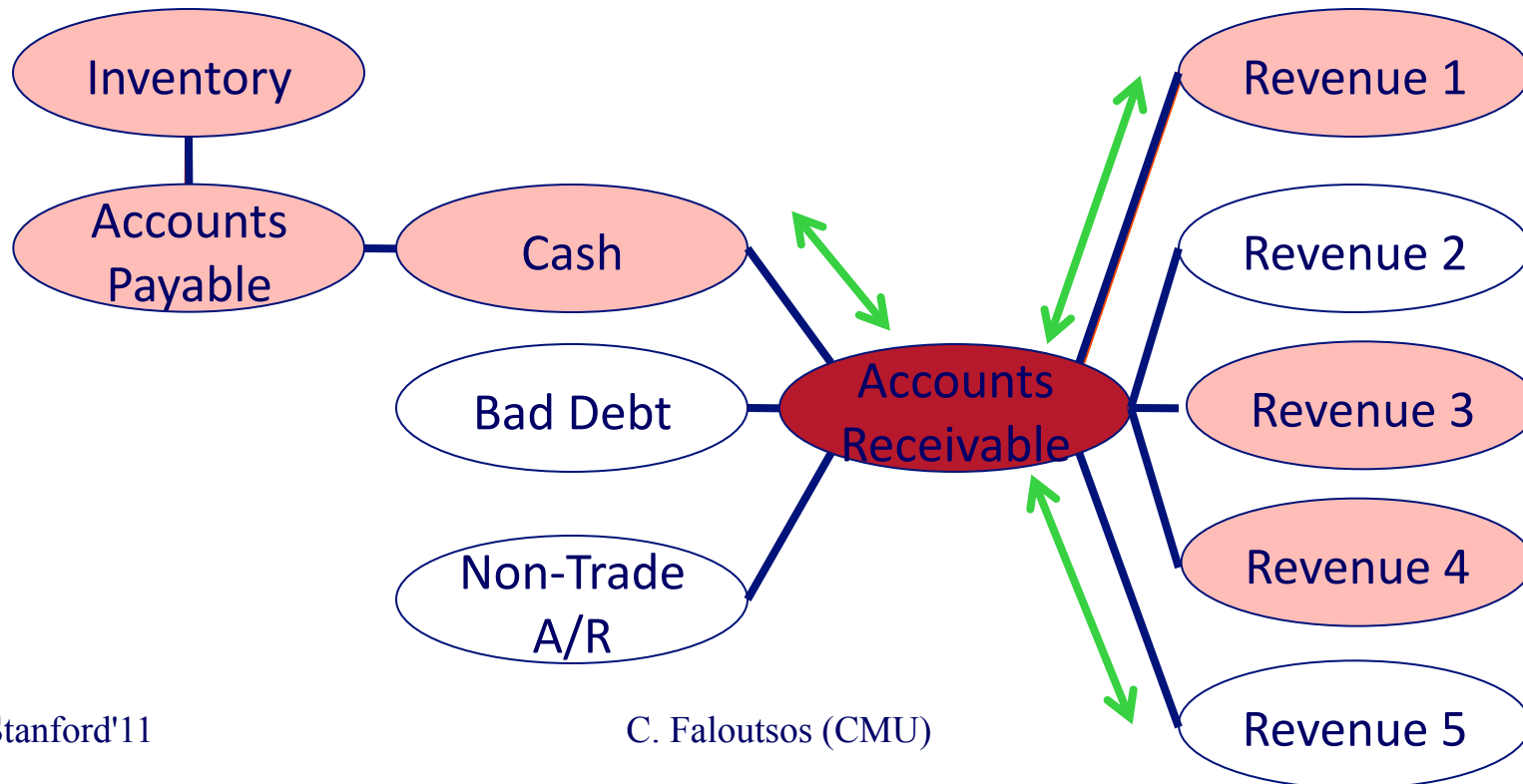
# Fraud detection

- **Problem:** Given network and noisy domain knowledge about suspicious nodes (flags), which nodes are most risky?



# Fraud detection

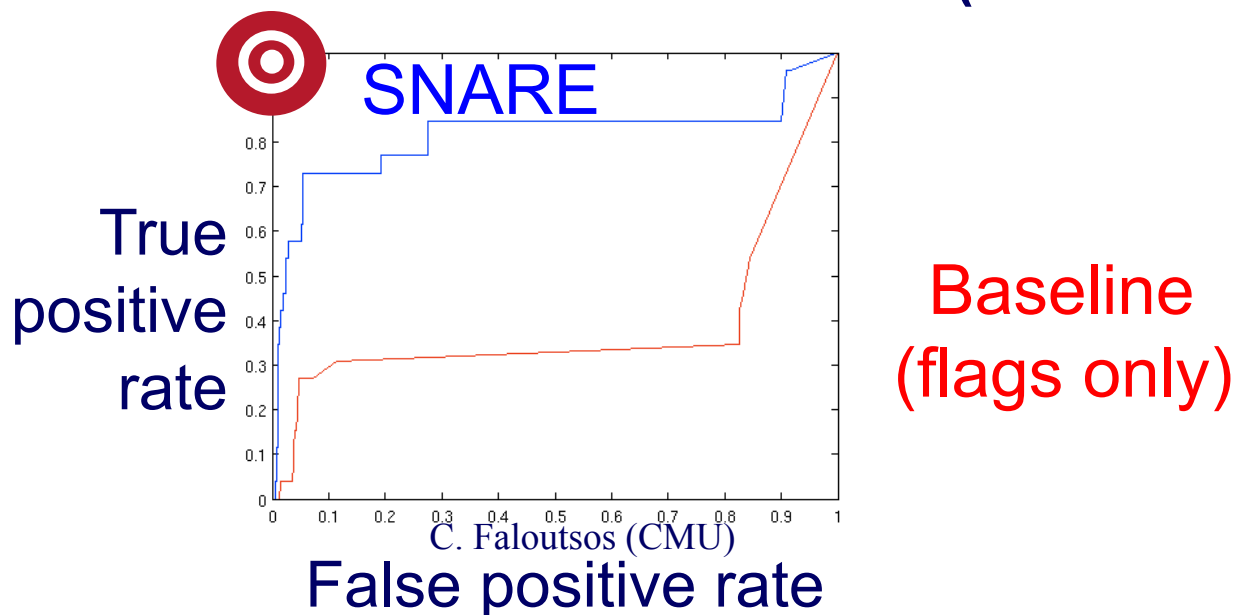
- **Problem:** Given network and noisy domain knowledge about suspicious nodes (flags), which nodes are most risky?



## BP and 'SNARE'

- **Accurate** – significant improvement over base
- **Flexible** - Can be applied to other domains
- **Scalable** - Linear time
- **Robust** - Works on large range of parameters

### Results for accounts data (ROC Curve)



# How to do B.P. on large graphs?

A: [U Kang, Polo Chau, +, ICDE'11],  
to appear

# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - OddBall (anomaly detection)
  - Belief propagation
  - ➔ – Immunization
- Problem#3: Scalability -PEGASUS
- Conclusions

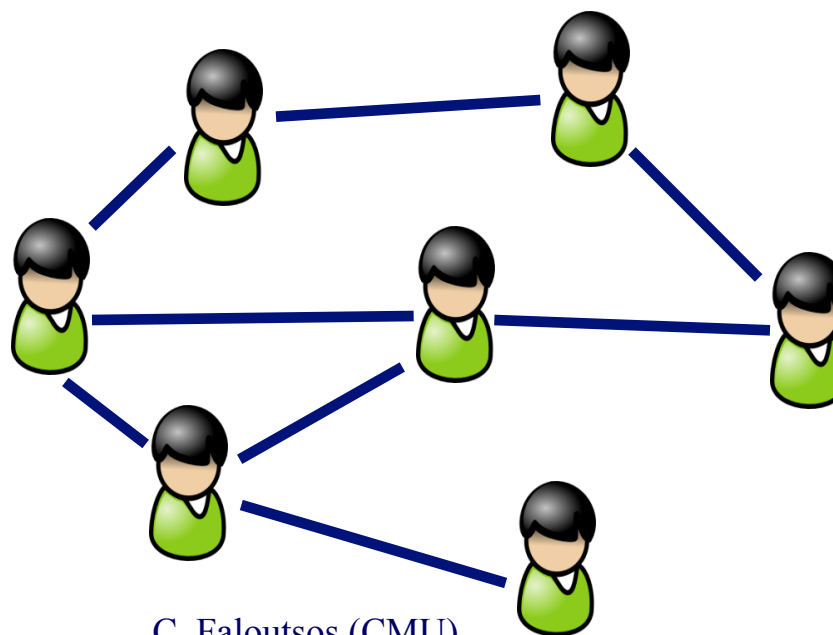


# Immunization and epidemic thresholds

- Q1: which nodes to immunize?
- Q2: will a virus vanish, or will it create an epidemic?

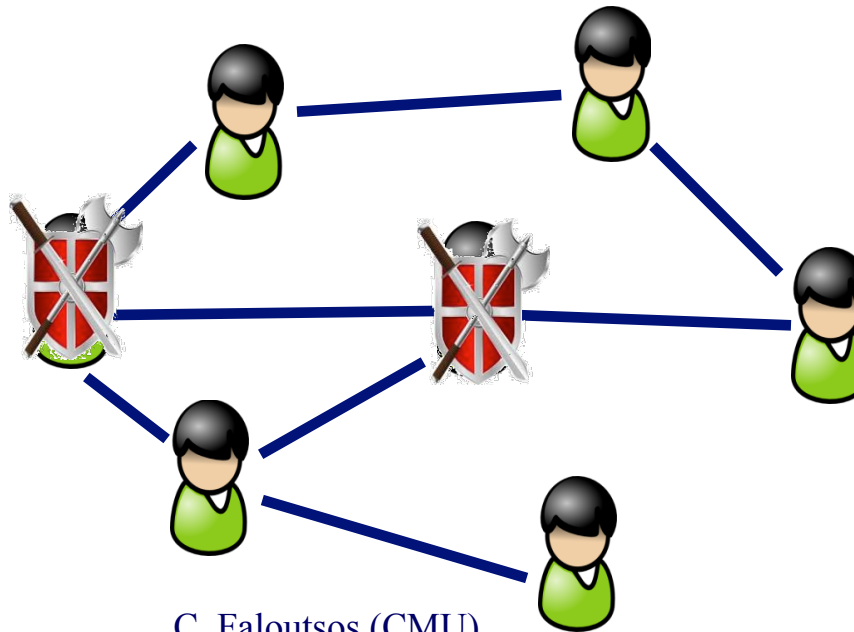
# Q1: Immunization:

- Given
  - a network,
  - k vaccines, and
  - the virus details
- Which nodes to immunize?



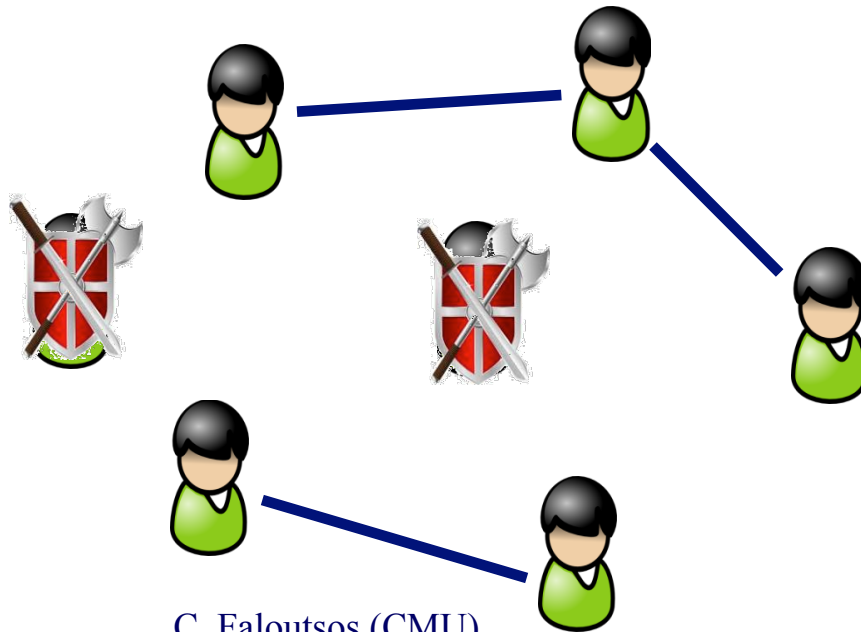
# Q1: Immunization:

- Given
  - a network,
  - k vaccines, and
  - the virus details
- Which nodes to immunize?



# Q1: Immunization:

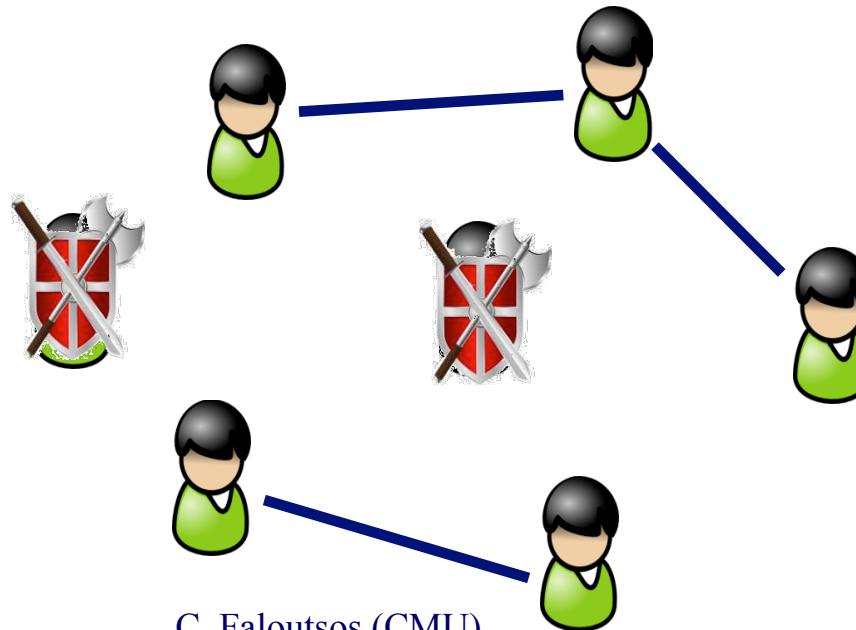
- Given
  - a network,
  - k vaccines, and
  - the virus details
- Which nodes to immunize?



# Q1: Immunization:

- Given
  - a network,
  - k vaccines, and
  - the virus details
- Which nodes to immunize?

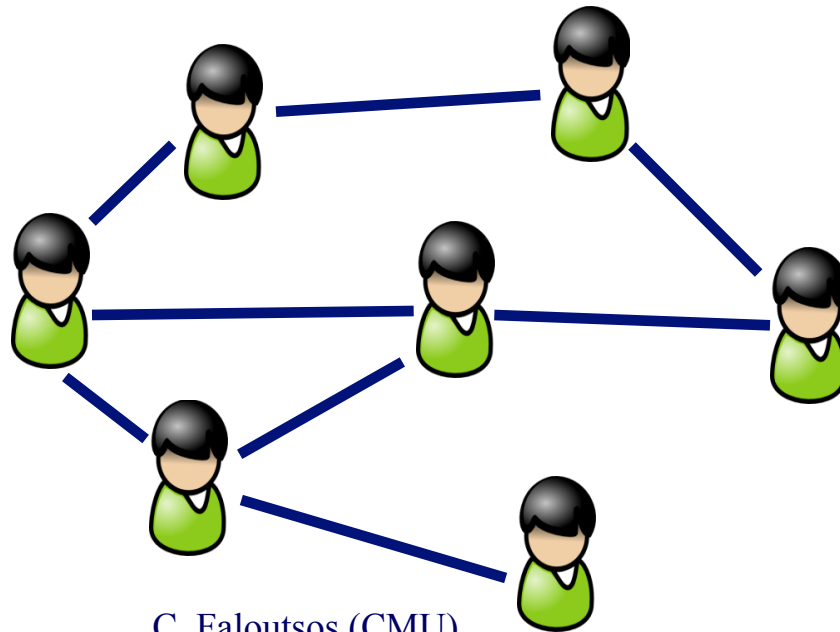
A: immunize the ones that maximally raise the 'epidemic threshold'  
[Tong+, ICDM'10]



## Q2: will a virus take over?

- Flu-like virus (no immunity, 'SIS')
- Mumps (life-time immunity, 'SIR')
- Pertussis (finite-length immunity, 'SIRS')

$\beta$ : attack prob  
 $\delta$ : heal prob



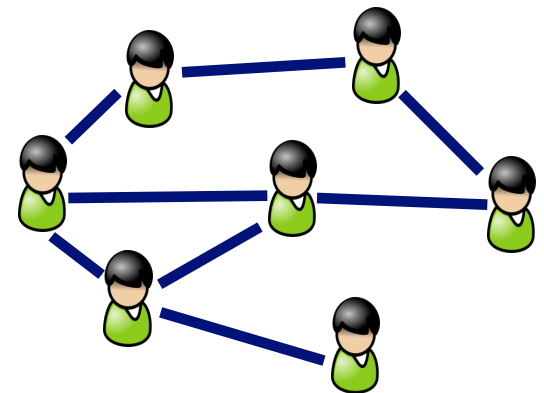
## Q2: will a virus take over?

- Flu-like virus (no immunity, 'SIS')
- Mumps (life-time immunity, 'SIR')
- Pertussis (finite-length immunity, 'SIRS')

$\beta$ : attack prob

$\delta$ : heal prob

A: depends on connectivity  
(avg degree? Max degree?  
variance? Something else?)



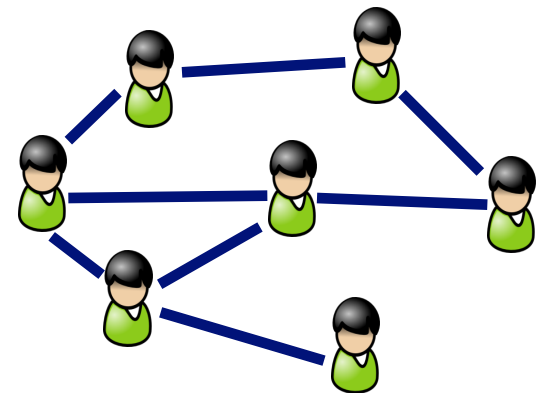
## Q2: will a virus take over?

- Flu-like virus (no immunity, ‘SIS’)
- Mumps (life-time immunity, ‘SIR’)
- Pertussis (finite-length immunity, ‘SIRS’)

$\beta$ : attack prob

$\delta$ : heal prob

A: depends on connectivity:  
ONLY on first eigenvalue





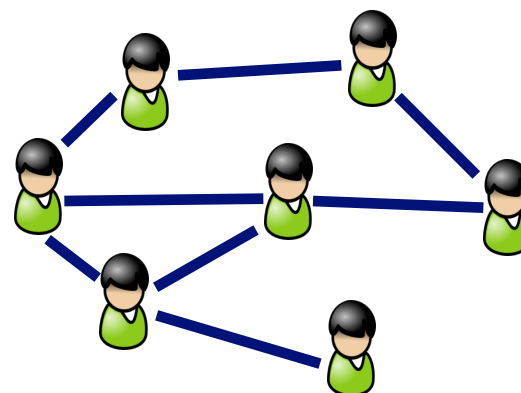
## A2: will a virus take over?

- For **all** typical virus propagation models (flu, mumps, pertussis, HIV, etc)
- The **only** connectivity easure that matters, is

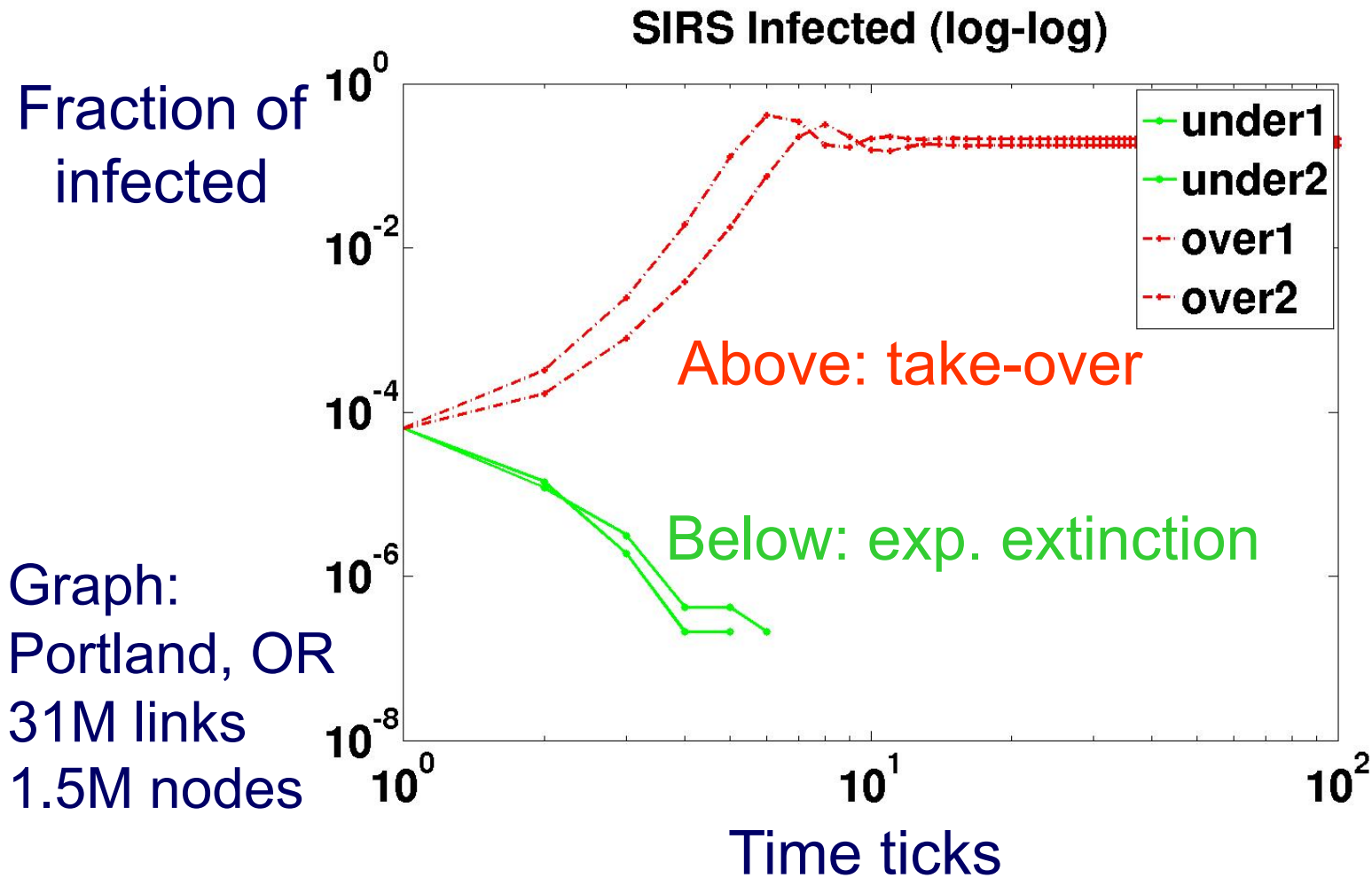
$$1/\lambda_1$$

the first eigenvalue of the  
adj. matrix

[Prakash+, arxiv]



# A2: will a virus take over?



# Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - OddBall (anomaly detection)
  - Belief propagation
  - Immunization
- ➔ • Problem#3: Scalability -PEGASUS
- Conclusions

# Scalability

- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, “*Web Search for a Planet: The Google Cluster Architecture*” IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?
- A: map/reduce – hadoop (open-source clone)  
<http://hadoop.apache.org/>



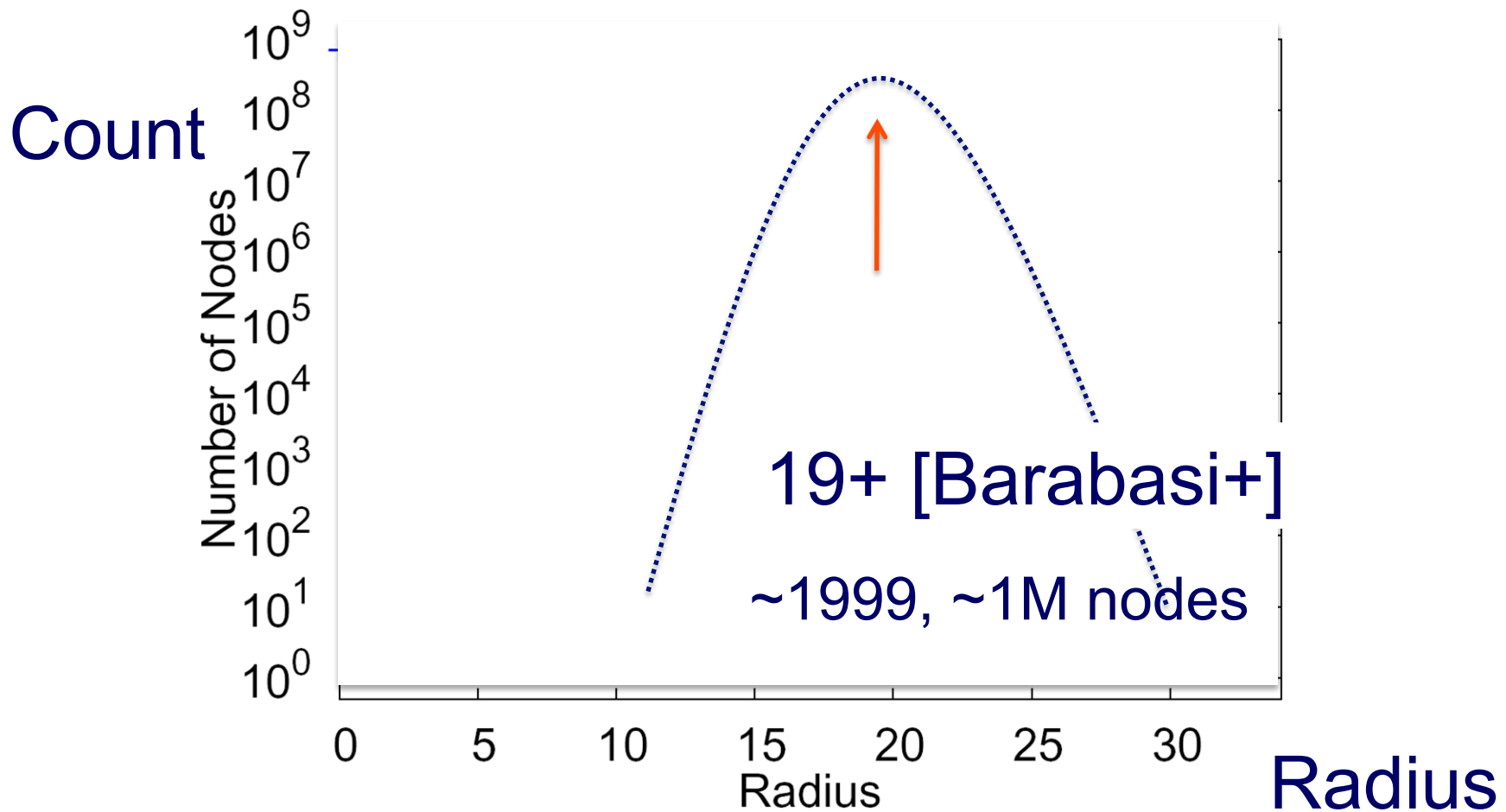
## Outline – Algorithms & results

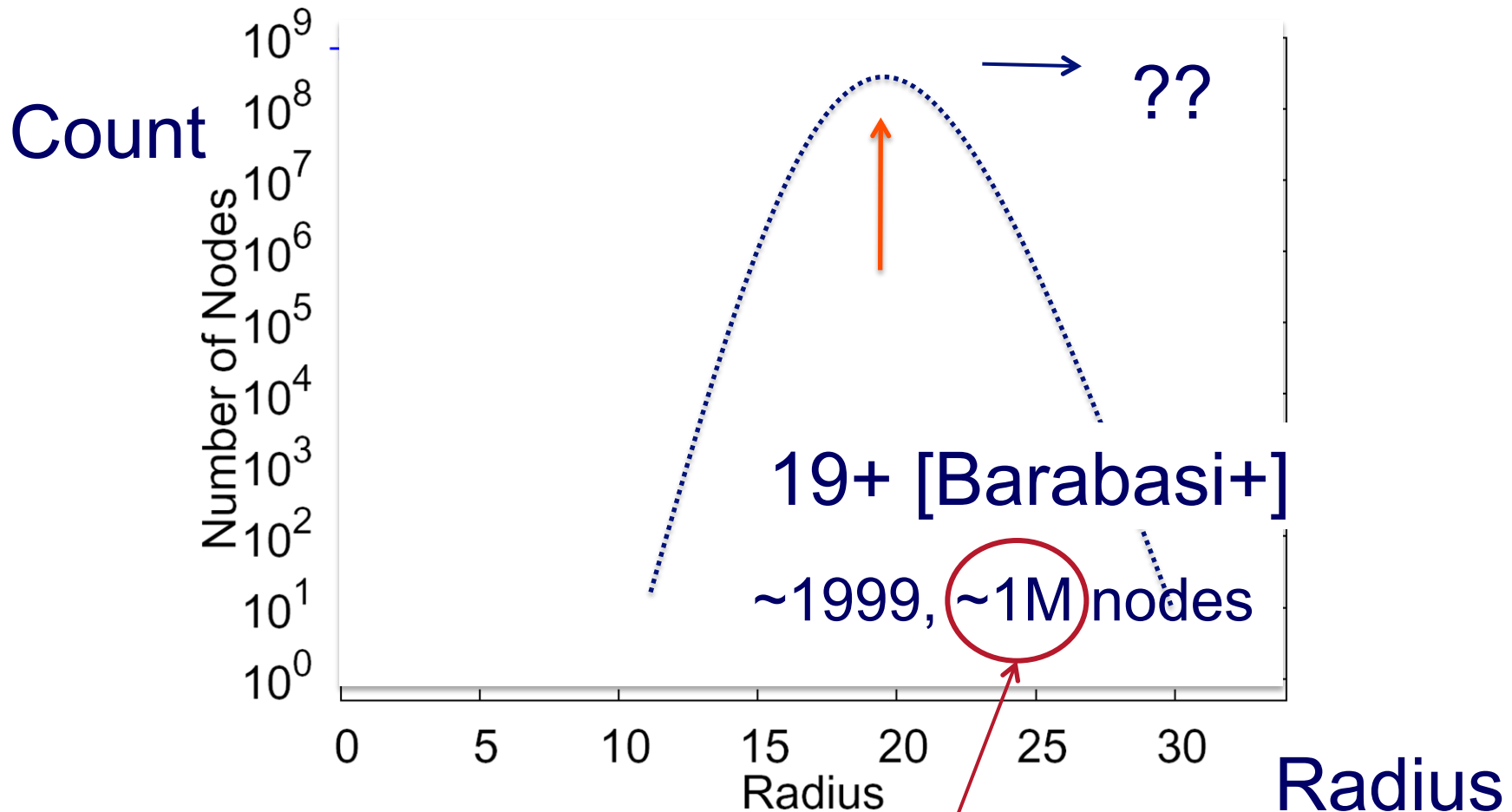
	<b>Centralized</b>	<b>Hadoop/ PEGASUS</b>
Degree Distr.	old	old
Pagerank	old	old
→ Diameter/ANF	old	<b>HERE</b>
Conn. Comp	old	<b>HERE</b>
Triangles	<b>done</b>	
Visualization	<b>started</b>	



## HADI for diameter estimation

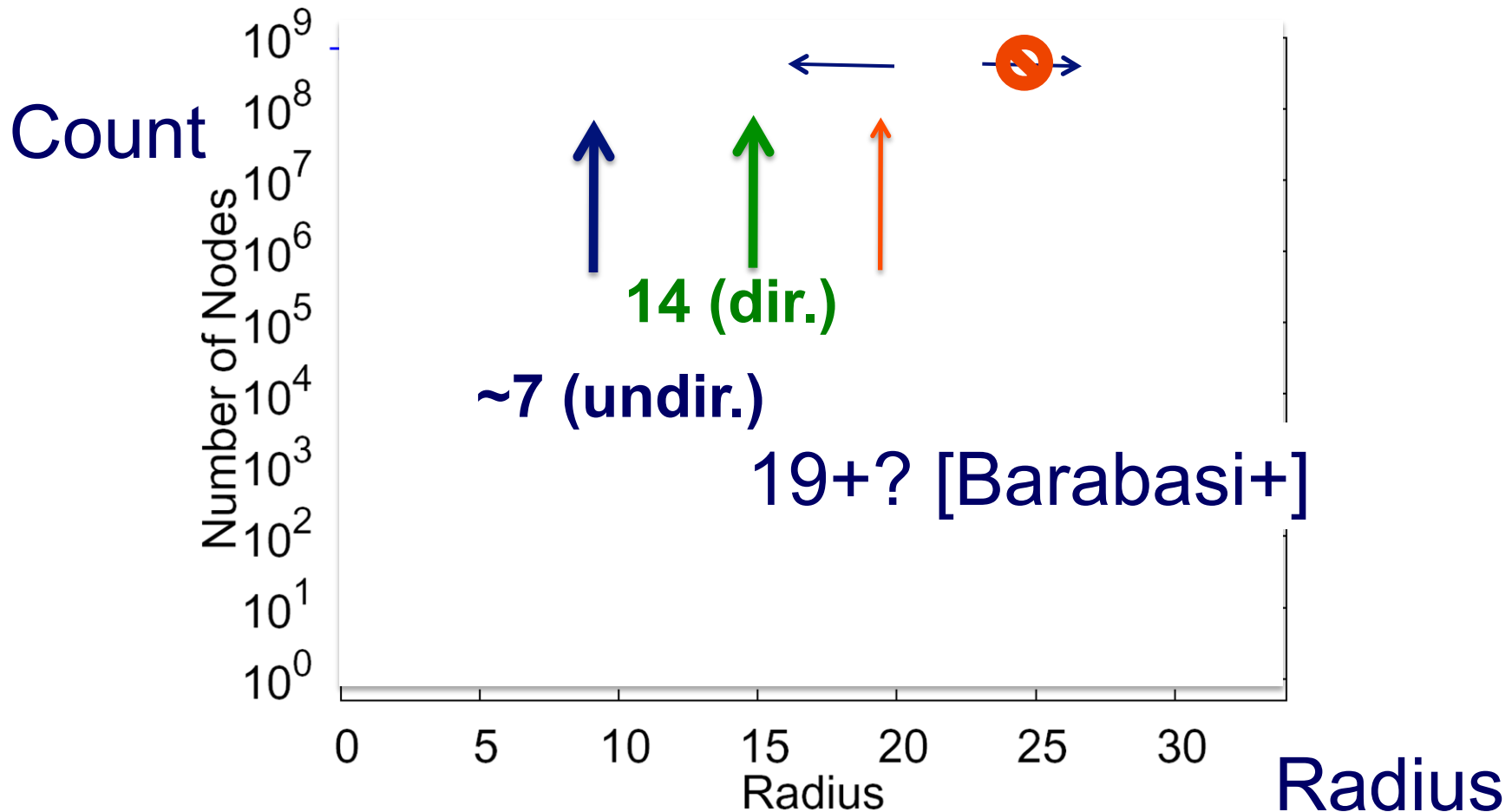
- *Radius Plots for Mining Tera-byte Scale Graphs* **U Kang**, Charalampos Tsourakakis, Ana Paula Appel, Christos Faloutsos, Jure Leskovec, SDM'10
- Naively: diameter needs  $O(N^2)$  space and up to  $O(N^3)$  time – **prohibitive** ( $N \sim 1B$ )
- Our HADI: linear on  $E$  ( $\sim 10B$ )
  - Near-linear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



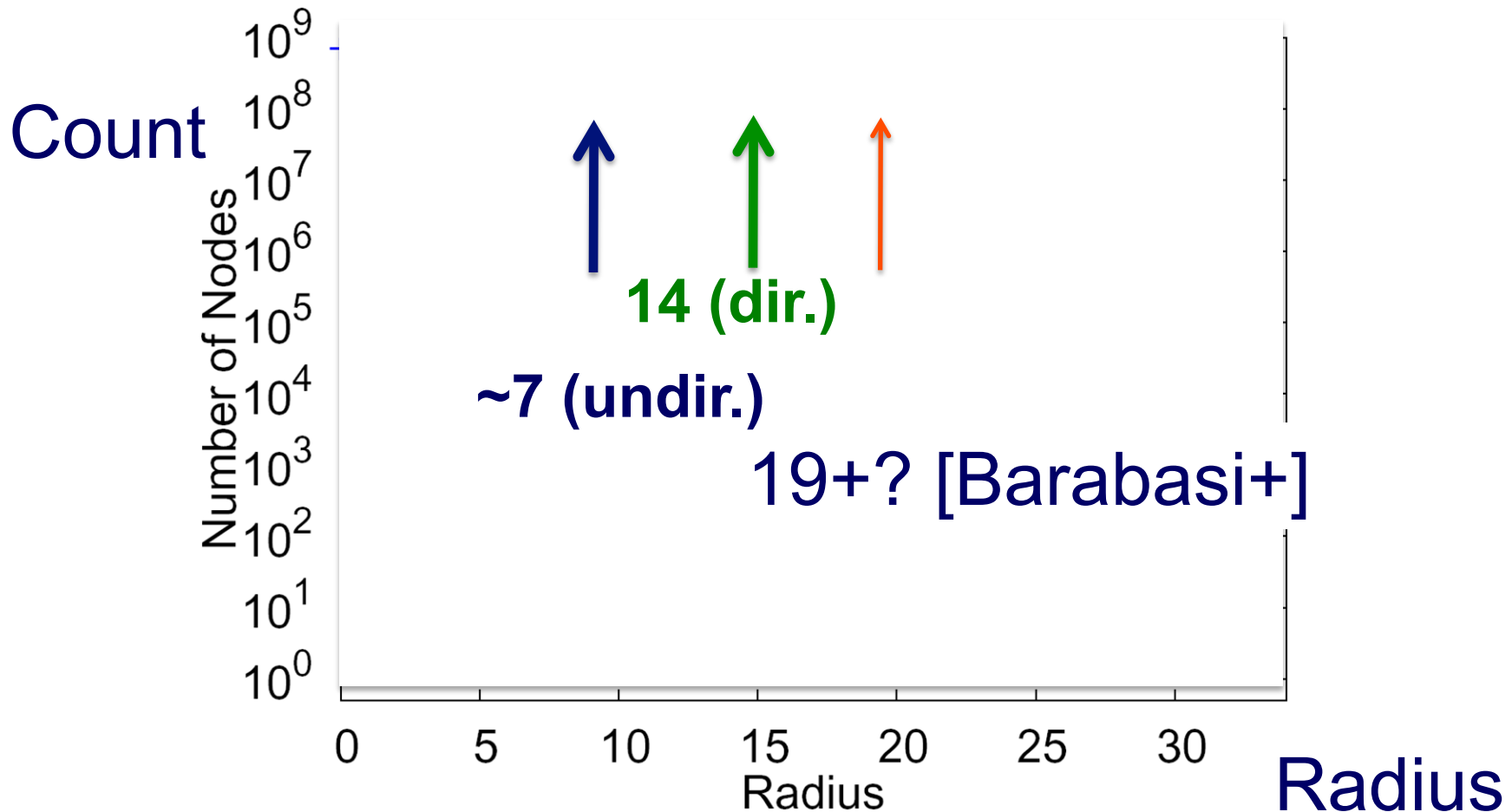


- YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- Largest publicly available graph ever studied.



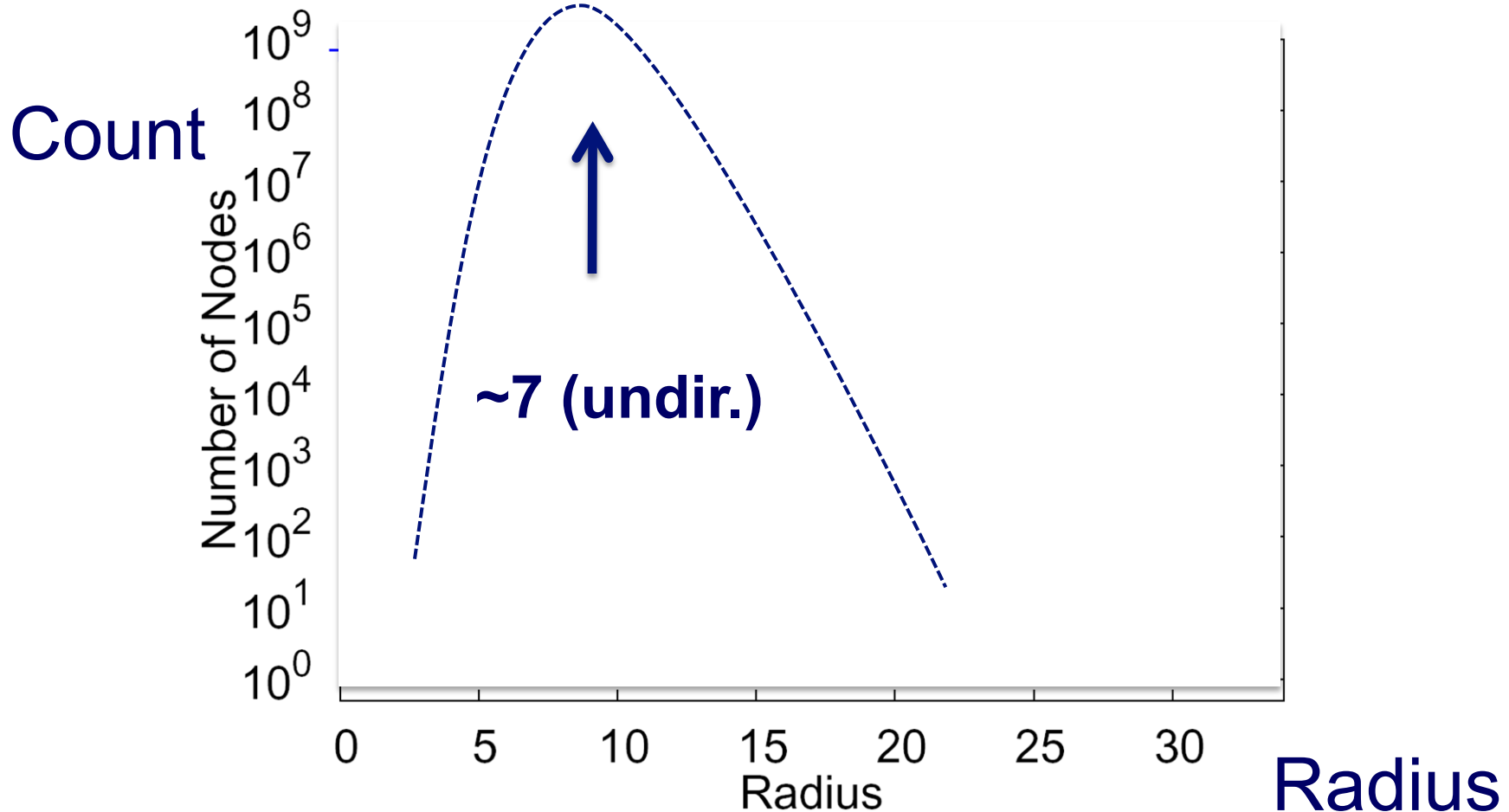


- YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- Largest publicly available graph ever studied.

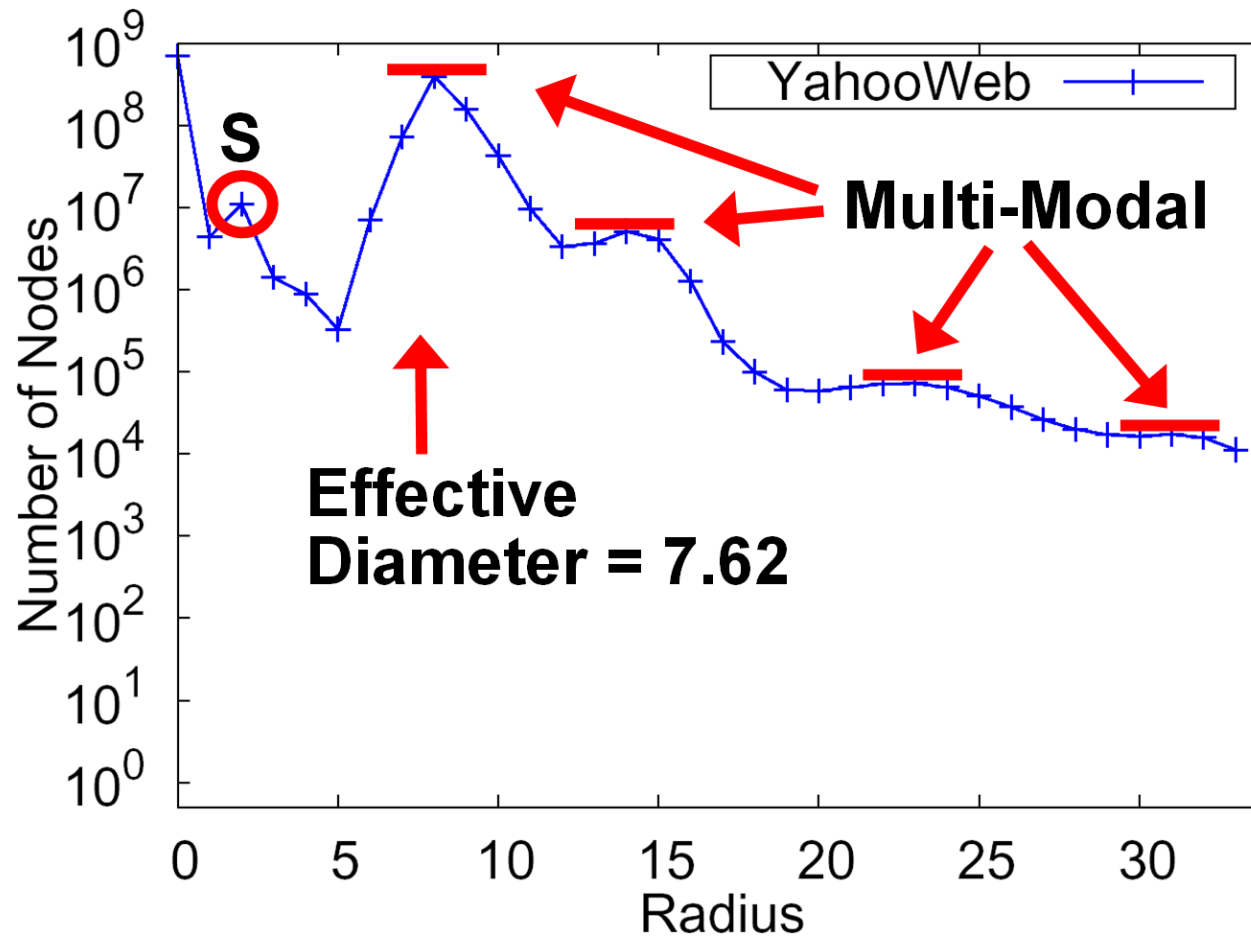


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

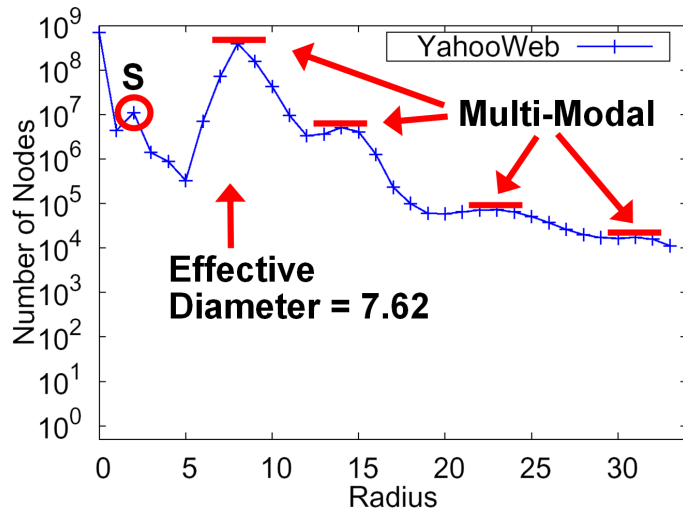


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)  
Q: Shape?

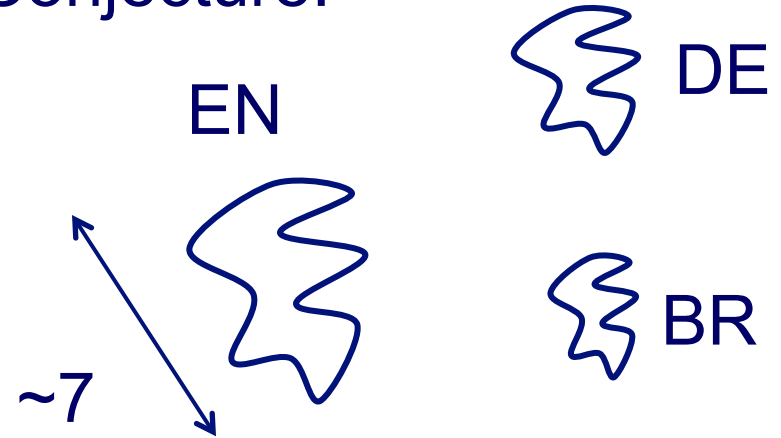


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

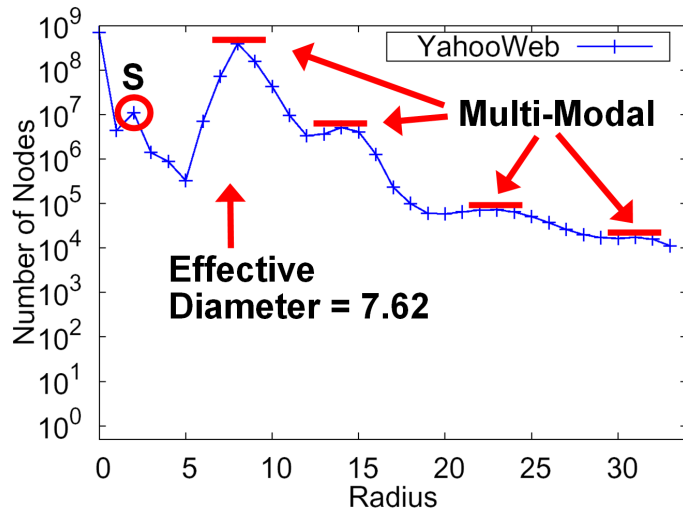


Conjecture:

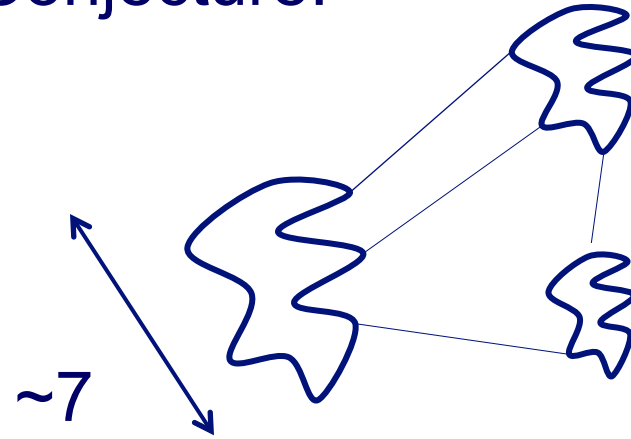


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

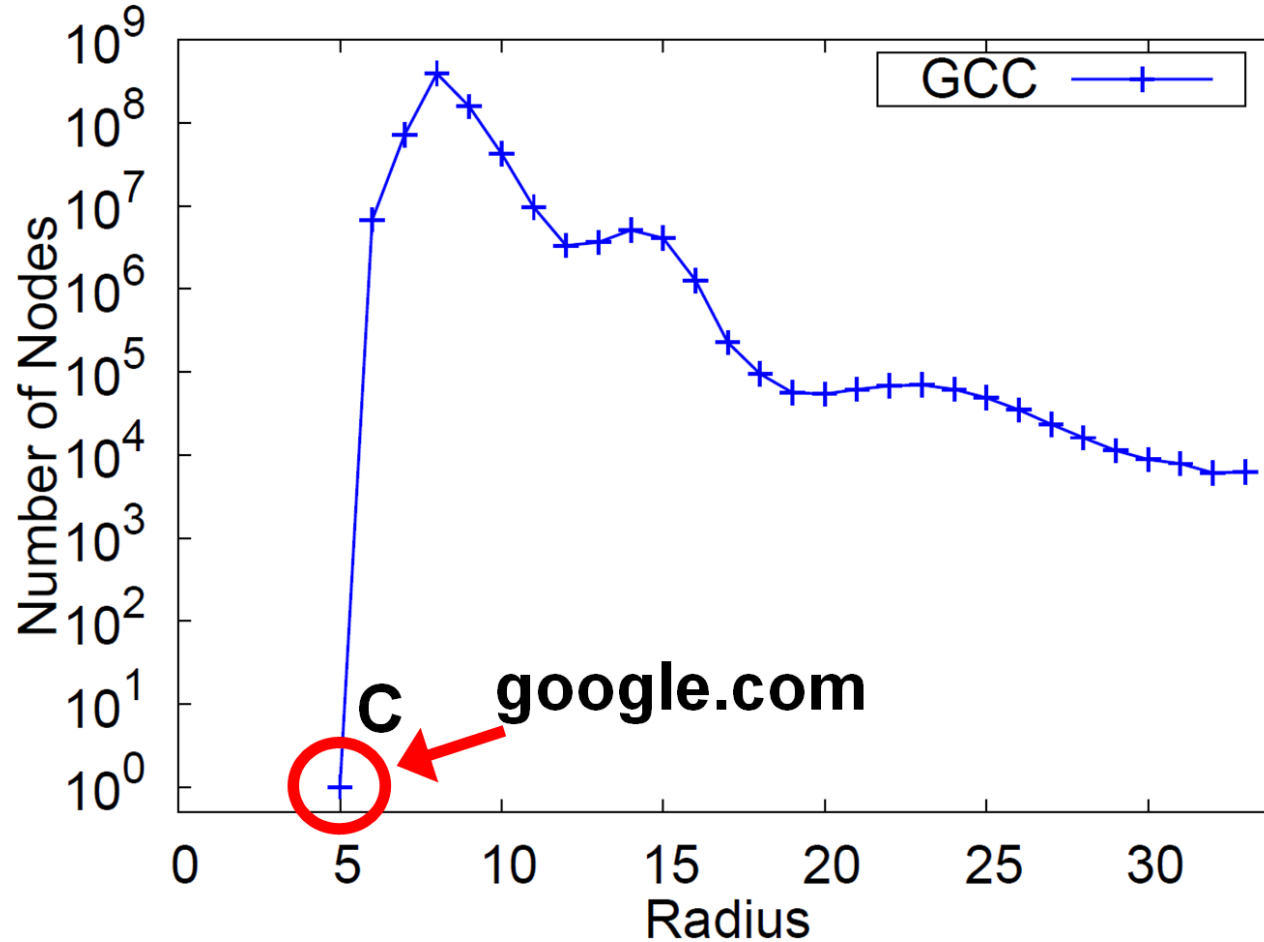
- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores .



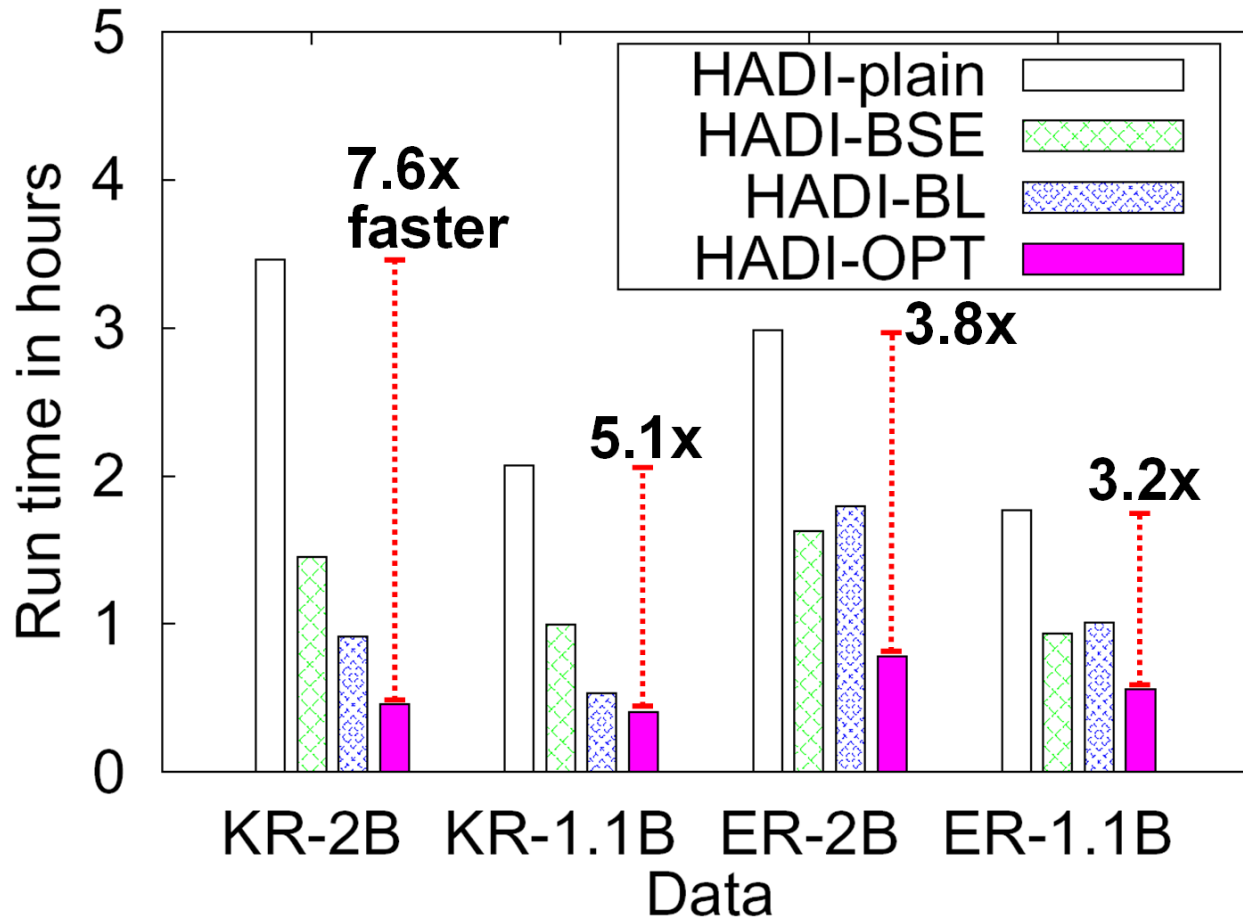
Conjecture:



- YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- effective diameter: surprisingly small.
  - Multi-modality: probably mixture of cores .



## Radius Plot of **GCC** of YahooWeb.



Running time - Kronecker and Erdos-Renyi  
Graphs with billions edges.



## Outline – Algorithms & results

	Centralized	Hadoop/ PEGASUS
Degree Distr.	old	old
Pagerank	old	old
Diameter/ANF	old	<b>HERE</b>
→ Conn. Comp	old	<b>HERE</b>
Triangles		done
Visualization	<b>started</b>	

# 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).

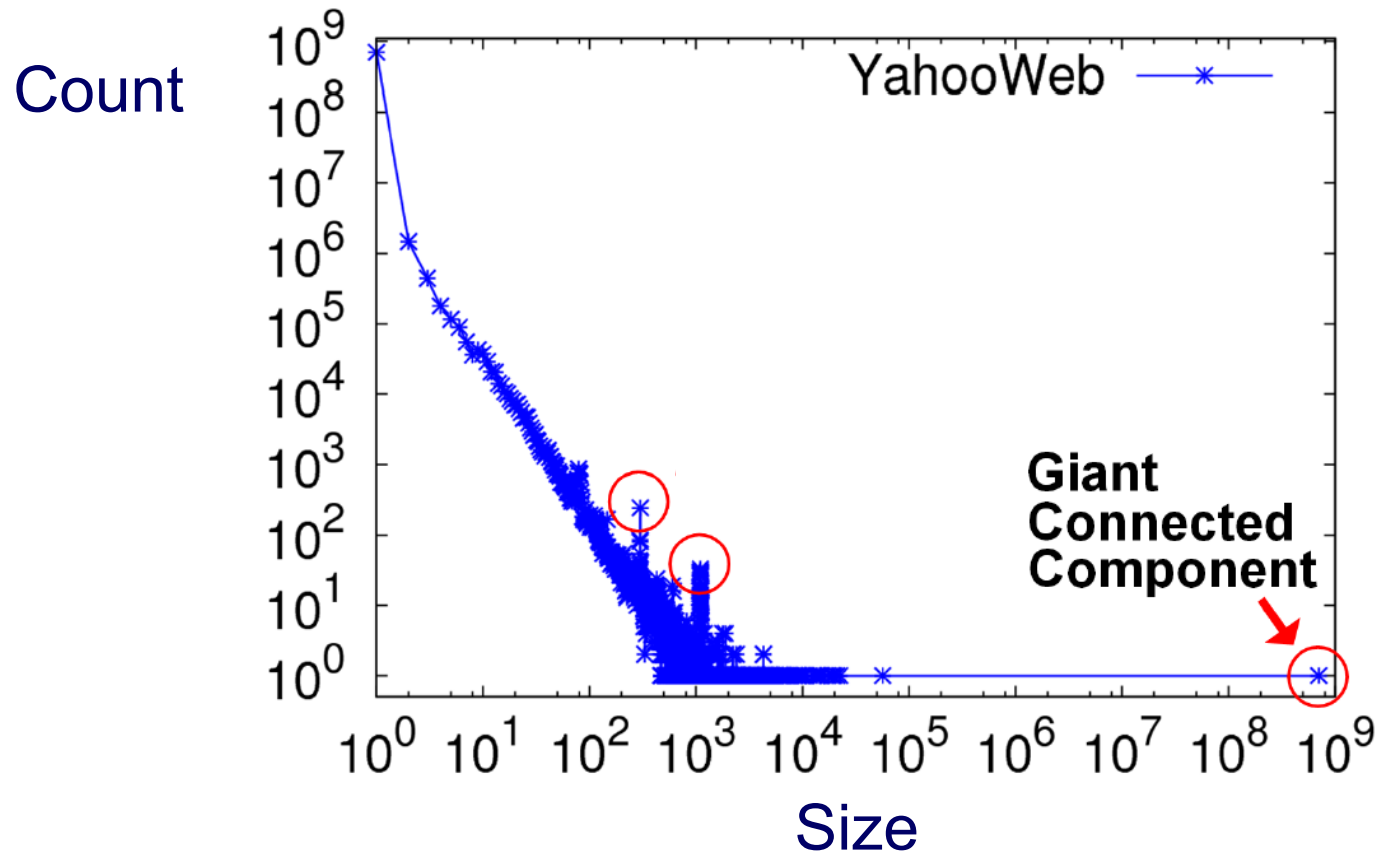
# Generalized Iterated Matrix Vector Multiplication (GIMV)

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

Matrix – vector  
Multiplication  
(iterated)

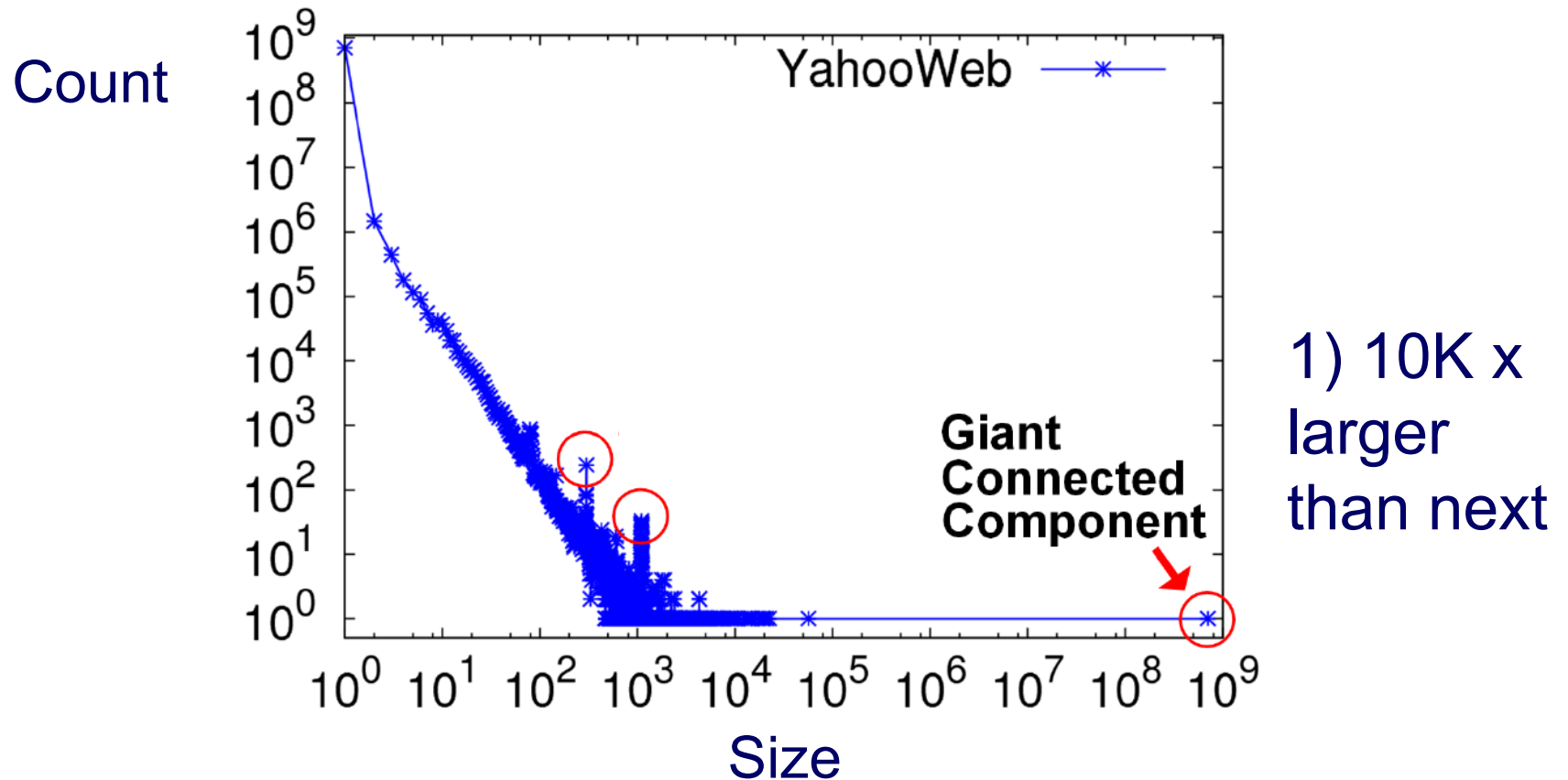
# Example: GIM-V At Work

- Connected Components – 4 observations:



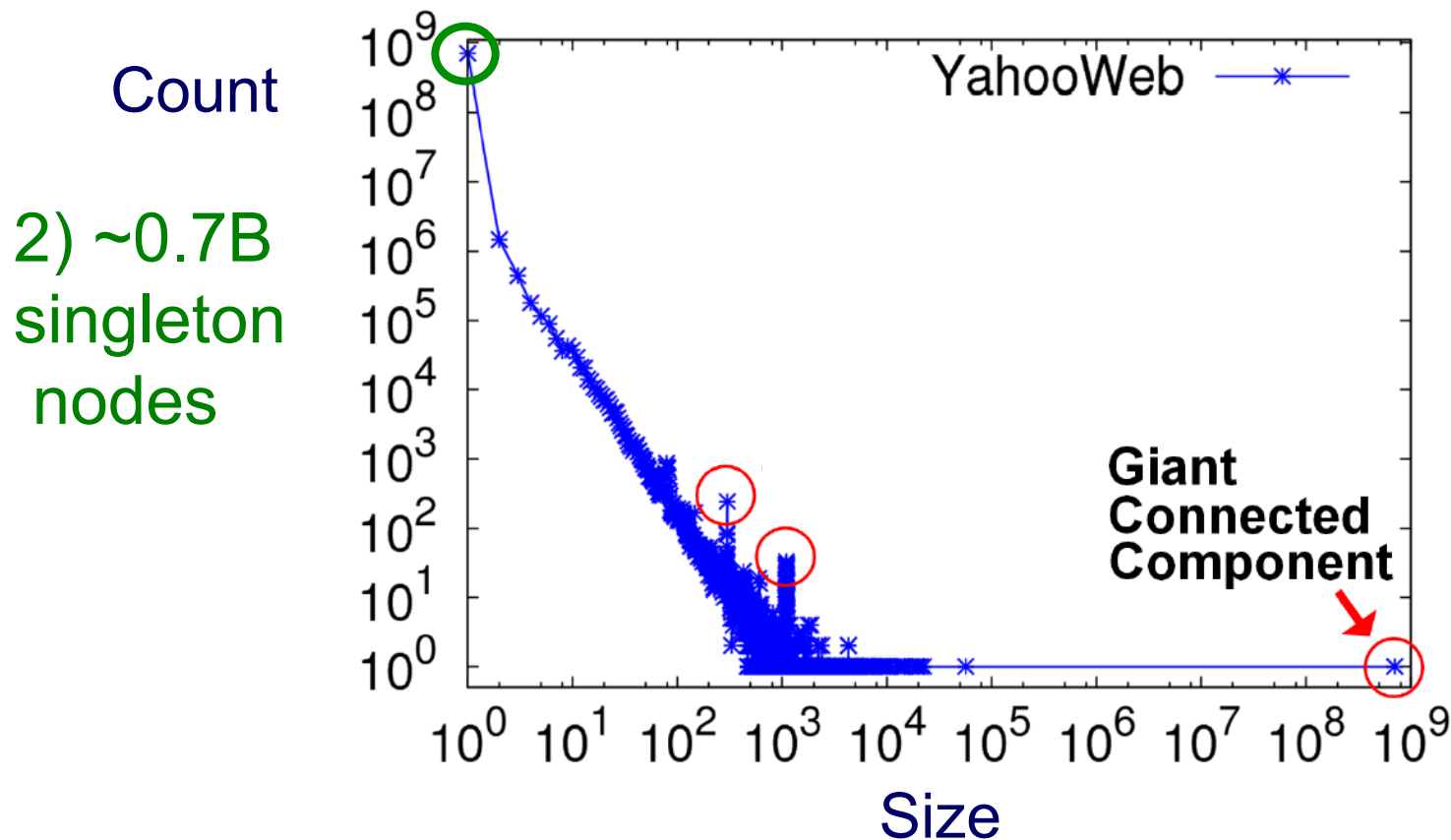
# Example: GIM-V At Work

- Connected Components



# Example: GIM-V At Work

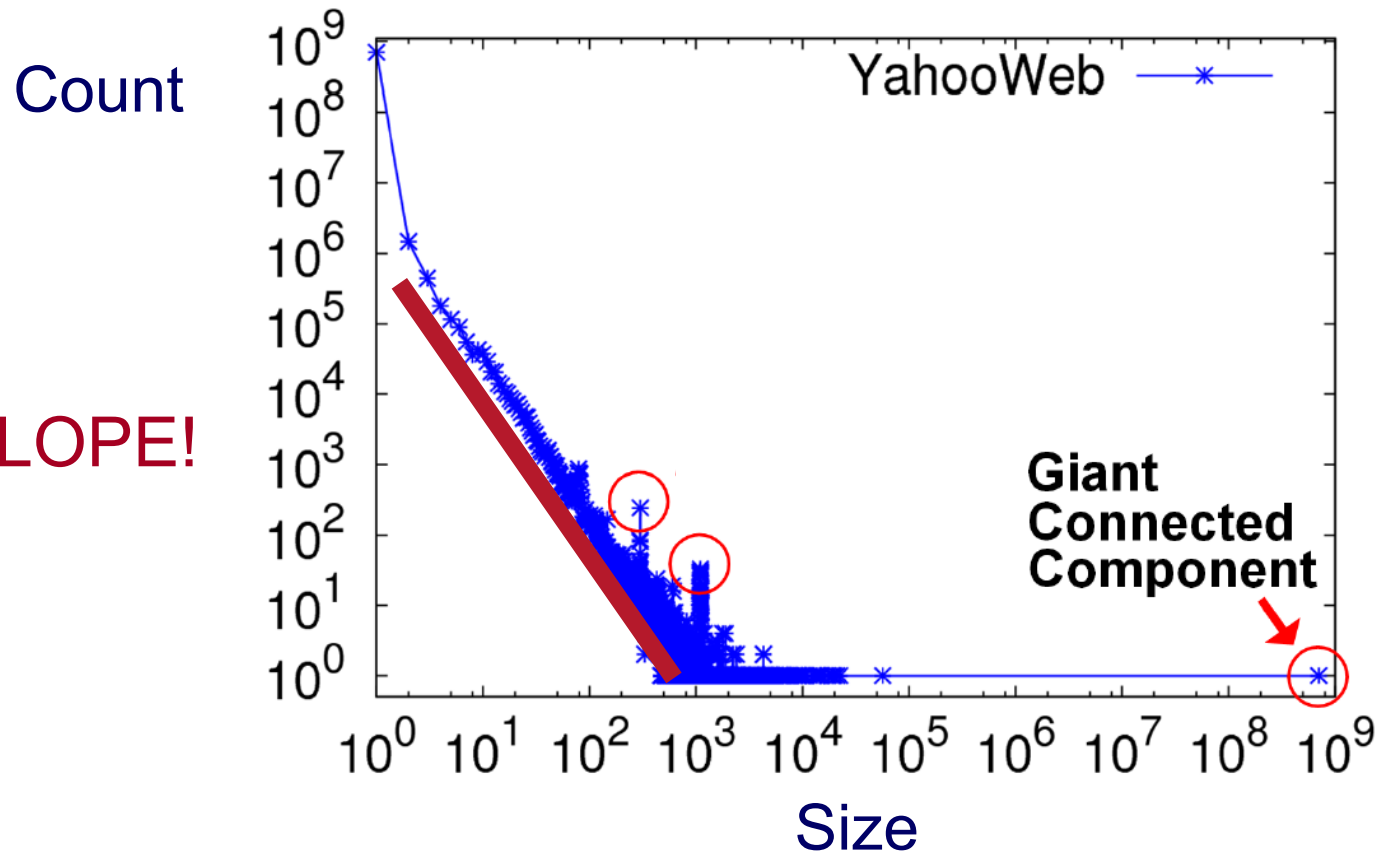
- Connected Components



# Example: GIM-V At Work

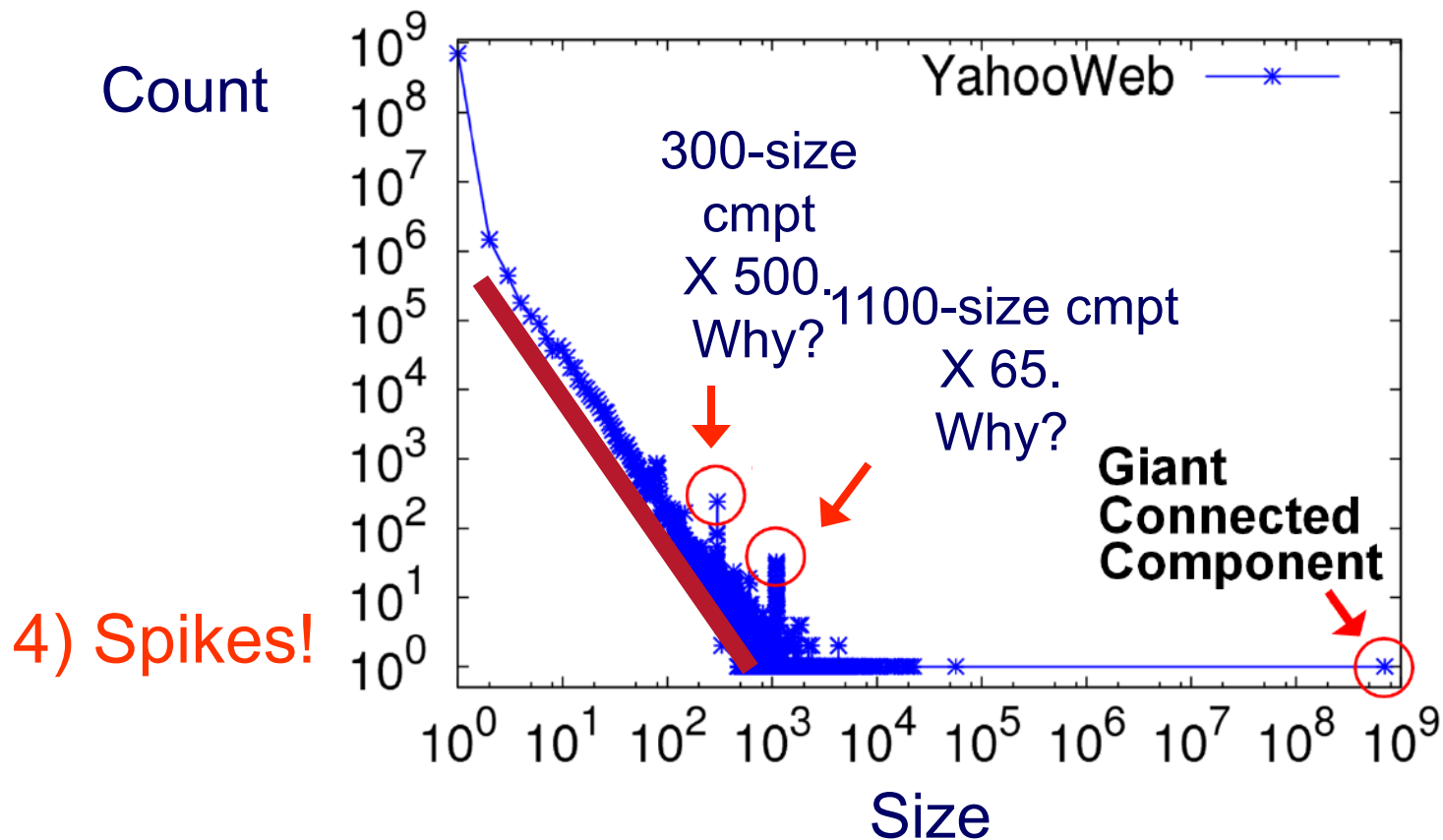
- Connected Components

3) SLOPE!



# Example: GIM-V At Work

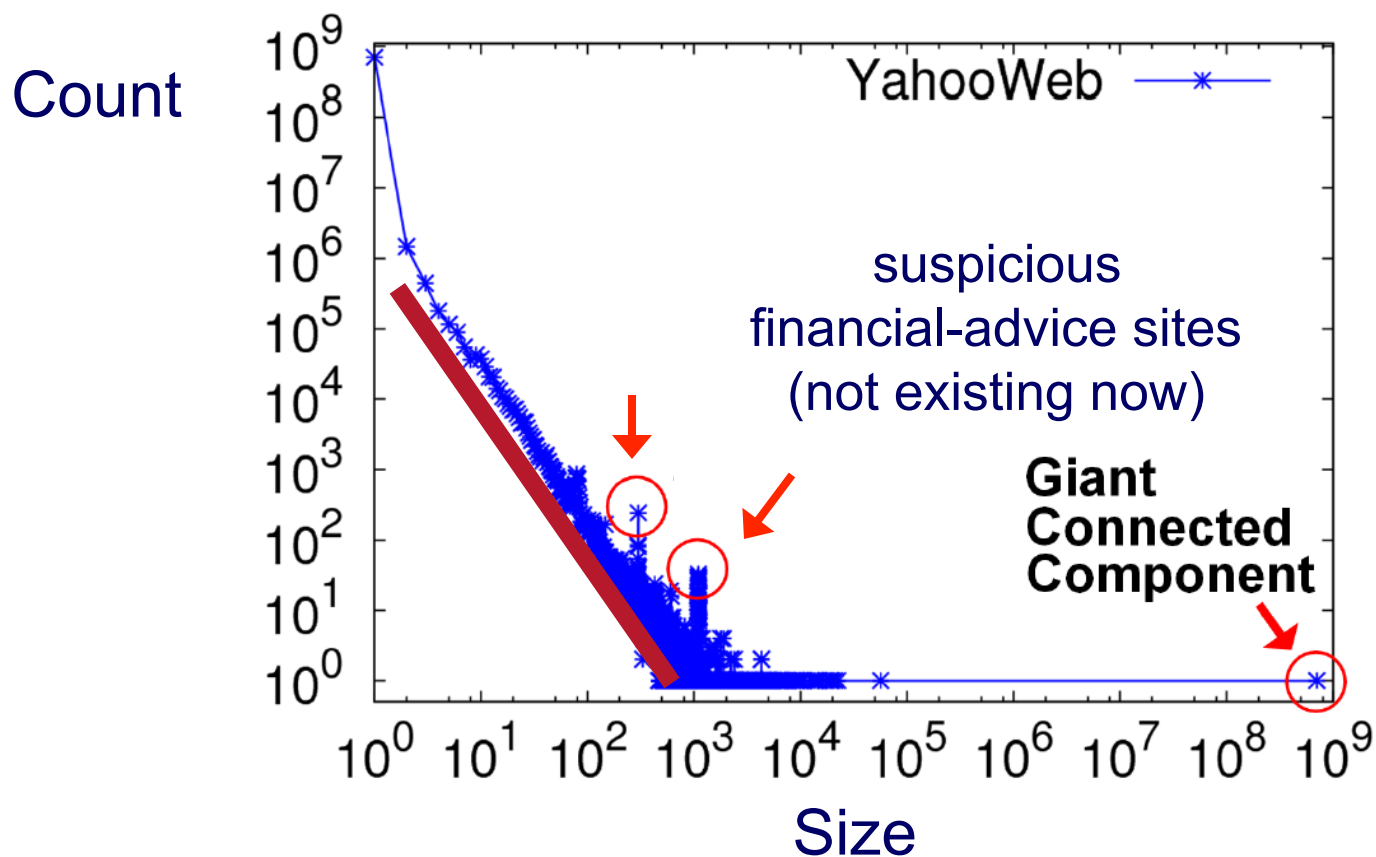
- Connected Components





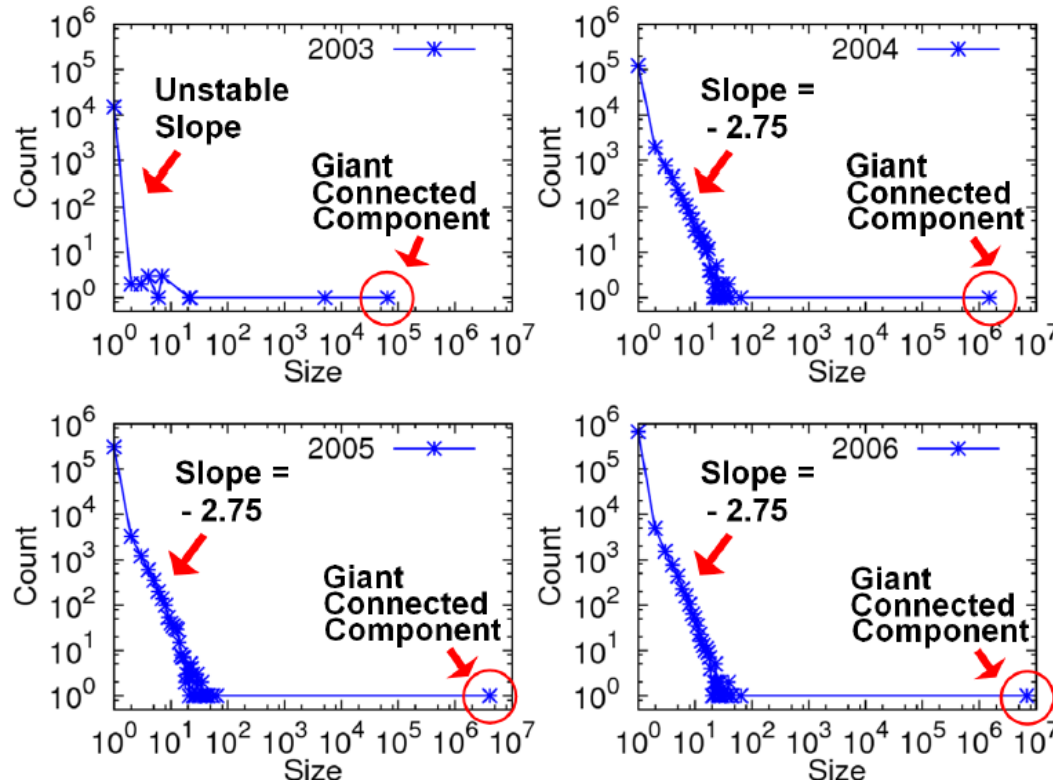
# Example: GIM-V At Work

- Connected Components



# GIM-V At Work

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



Stable tail slope  
after the gelling point

# Outline

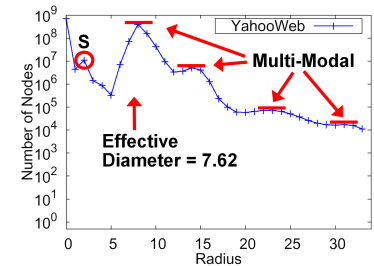
- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
- Problem#3: Scalability
- ➔ • Conclusions

# OVERALL CONCLUSIONS – low level:

- Several new **patterns** (fortification, triangle-laws, conn. components, etc)
- New **tools**:
  - anomaly detection (OddBall), belief propagation, immunization
- **Scalability**: PEGASUS / hadoop

# OVERALL CONCLUSIONS – high level

- Large datasets reveal patterns/outliers that are invisible otherwise
- Terrific opportunities
  - Large datasets, easily(\*) available PLUS
  - s/w and h/w developments



# 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.

# 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](http://www.cs.cmu.edu/~pegasus)



Chau,  
Polo



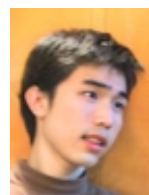
McGlohon,  
Mary



Tong,  
Hanghang



Akoglu,  
Leman



Kang, U



Prakash,  
Aditya



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