Carnegie Mellon

Mining Billion-Node Graphs - Patterns and Algorithms

Christos Faloutsos

CMU

Thank you!

- Panos Chrysanthis
- Ling Liu
- Vladimir Zadorozhny
- Prashant Krishnamurthy

Resource

Open source system for mining huge graphs:

PEGASUS project (PEta GrAph mining System)

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

Carnegie Mellon

Roadmap

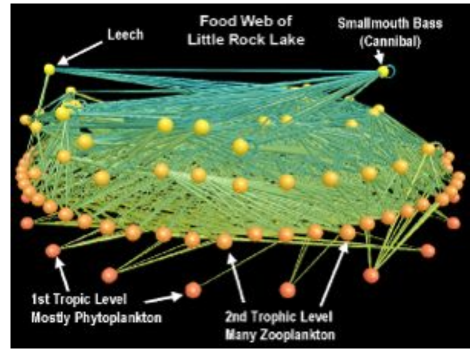
- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
- Problem#3: Scalability
- Conclusions

Graphs - why should we care?

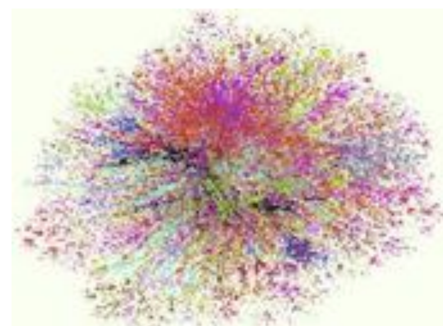
LinkedIn

>$10B revenue

>0.5B users



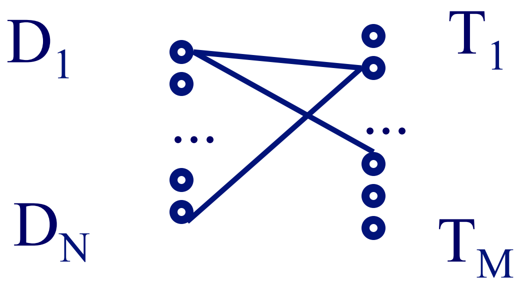
Food Web
[Martinez ’91]



Internet Map
[lumeta.com]

Graphs - why should we care?

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



- web: hyper-text graph

- ... and more:

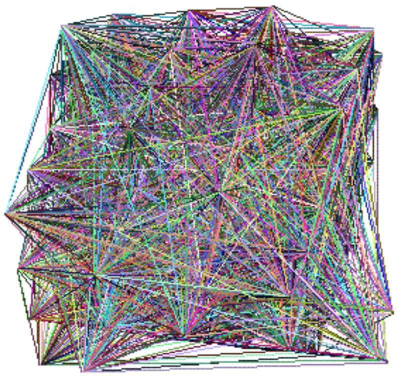
Graphs - why should we care?

- 'viral' marketing
- web-log ('blog') news propagation
- computer network security: email/IP traffic and anomaly detection
- ....
- Subject-verb-object -> graph
- Many-to-many db relationship -> graph

Outline

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

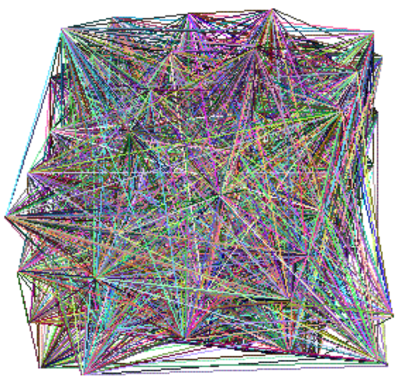
Problem #1 - network and graph mining



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

- What is 'normal'/'abnormal'?
- which patterns/laws hold?

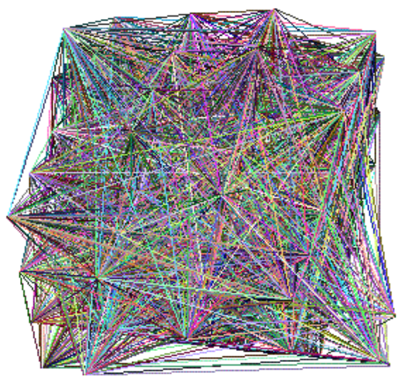
Problem #1 - network and graph mining



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

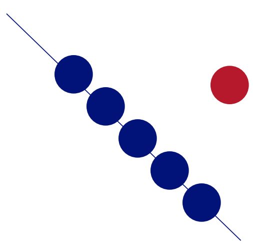
- What is 'normal'/'abnormal'?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**

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

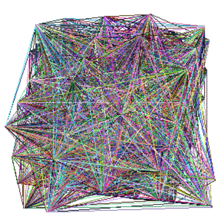
- Are real graphs random?
- A: NO!!
  - Diameter
  - in- and out- degree distributions
  - other (surprising) patterns

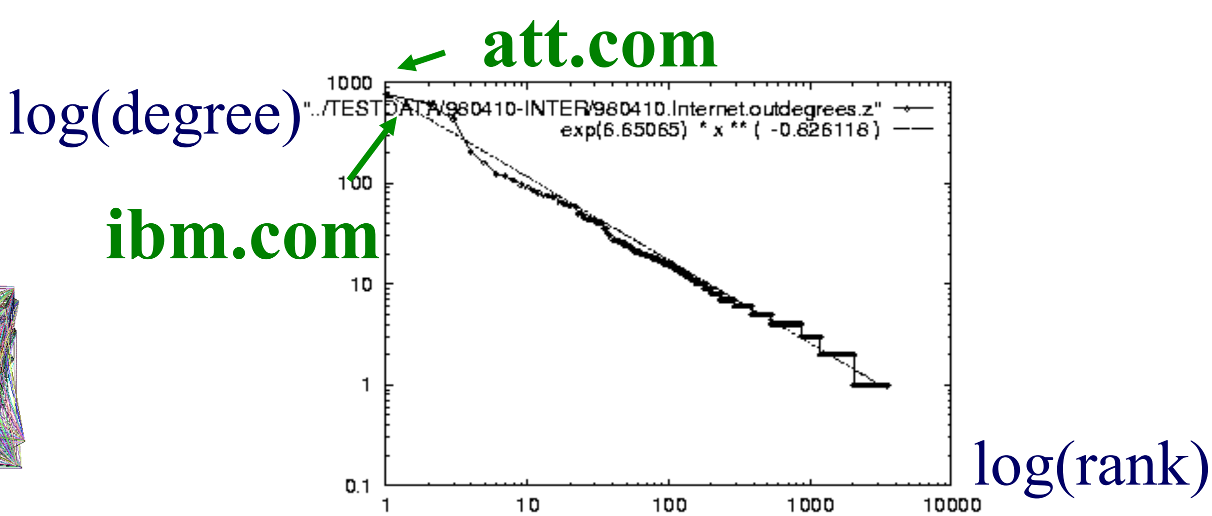
- So, let’s look at the data

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains



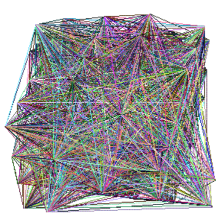


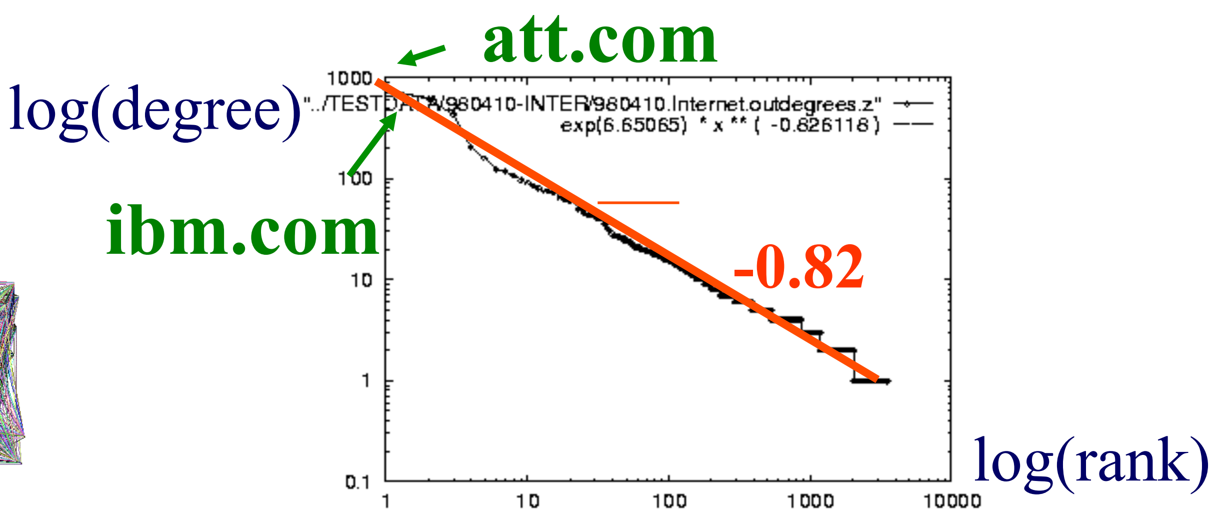
Carnegie Mellon

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

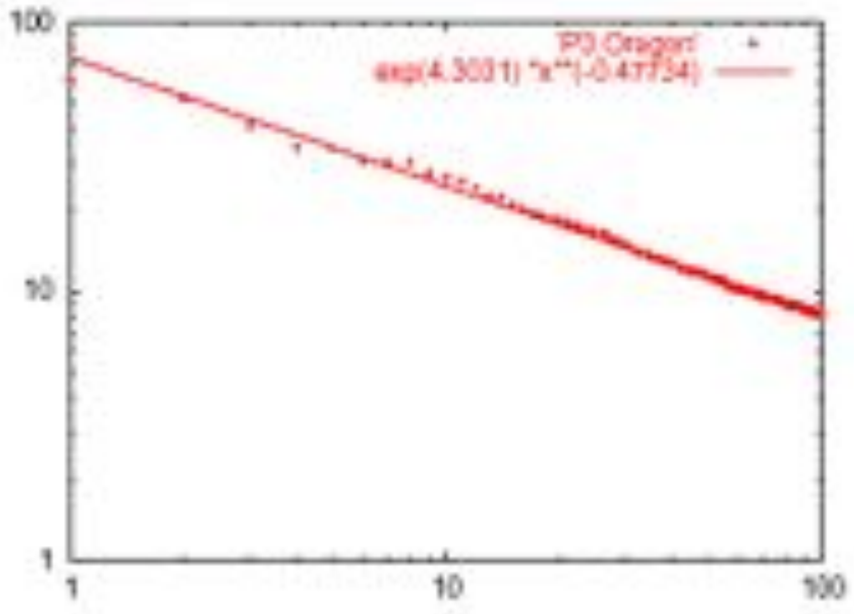




Carnegie Mellon

Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

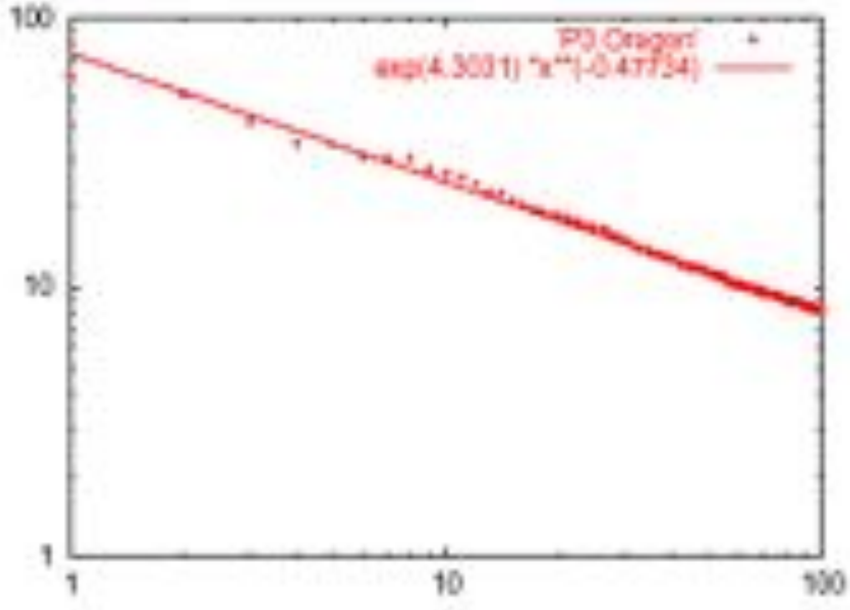
E = -0.48

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

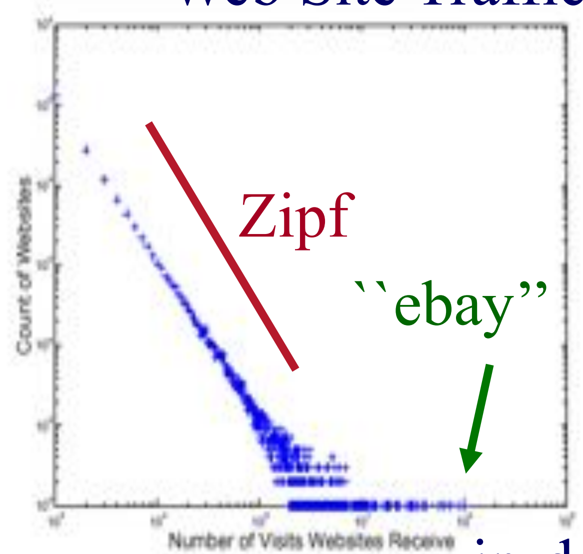
- [Mihail, Papadimitriou ’02]: slope is ½ of rank exponent

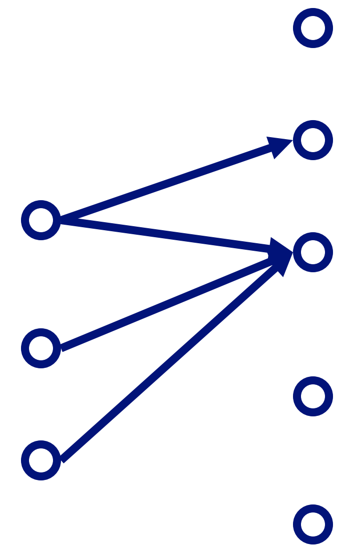
But:

How about graphs from other domains?

More power laws:

- web hit counts [w/ A. Montgomery]

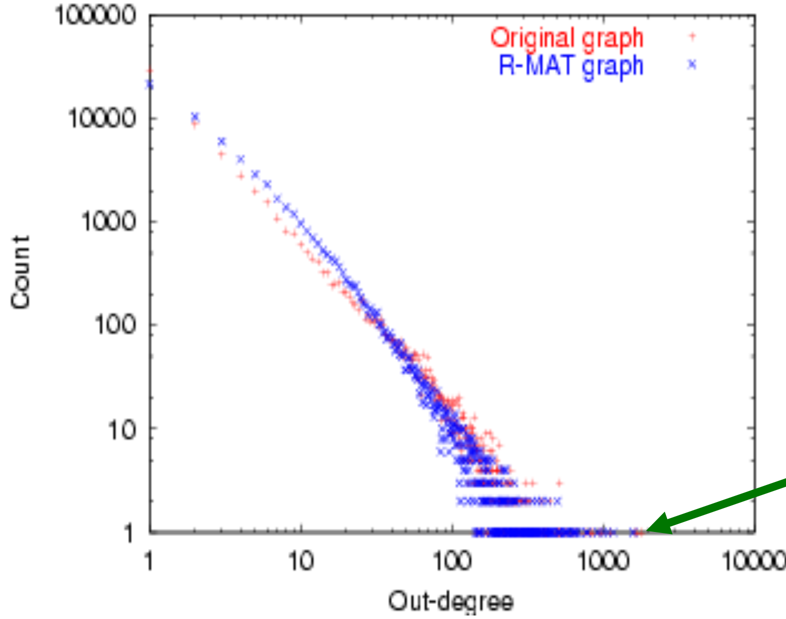




epinions.com

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

count



trusts-2000-people user

(out) degree

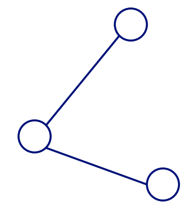
And numerous more

- # 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

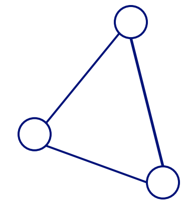
- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - triangles
    - cliques
  - Weighted graphs
  - Time evolving graphs
- Problem#2: Tools

Solution# S.3: Triangle 'Laws'



- Real social networks have a lot of triangles

Solution# S.3: Triangle ‘Laws’

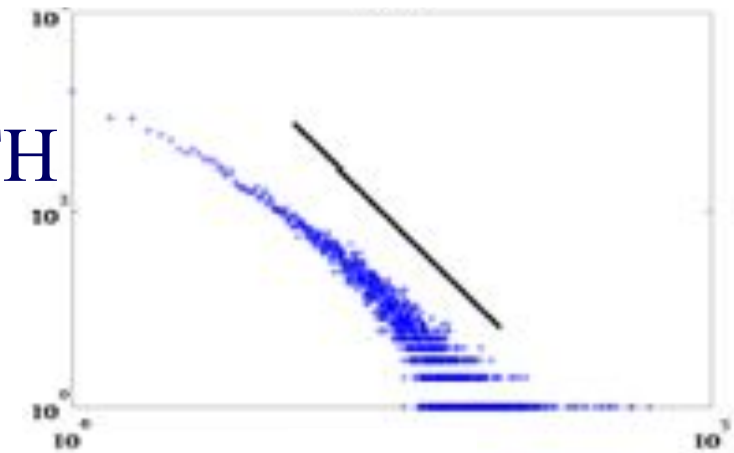


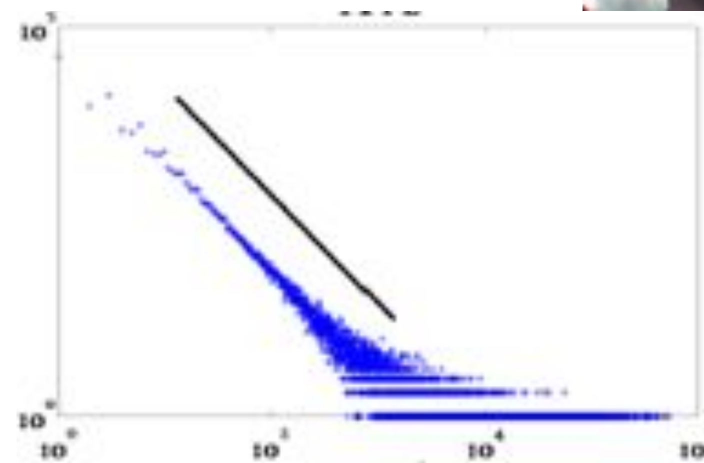
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

Carnegie Mellon

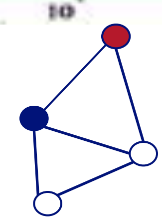
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

HEP-TH

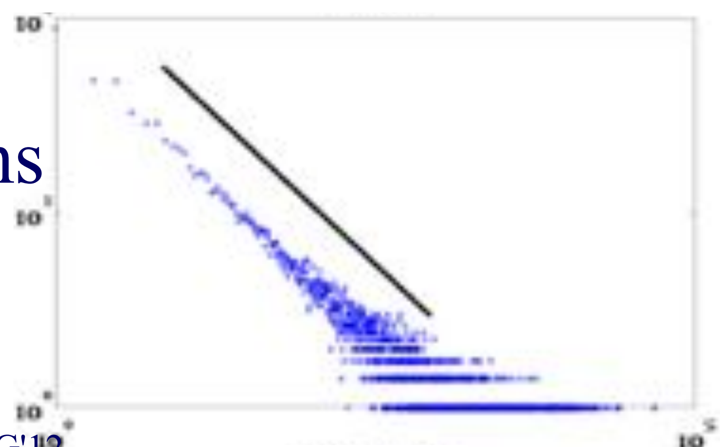




ASN



Epinions



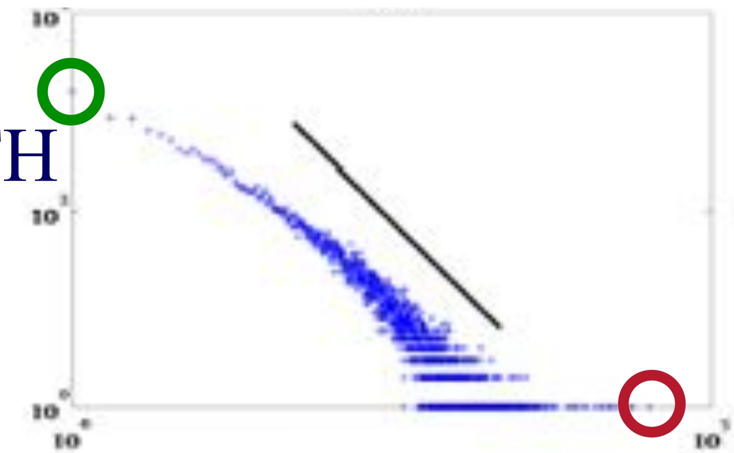
X-axis: # of participating triangles

Y: count (~ pdf)

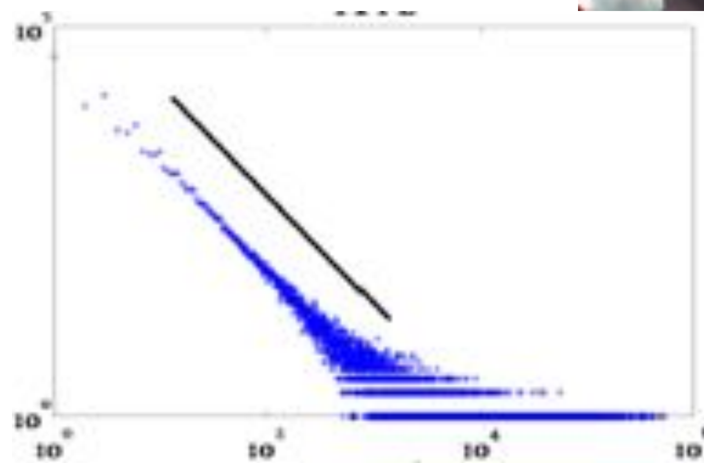
Carnegie Mellon

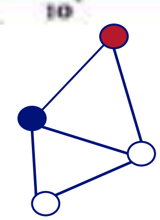
Triangle Law: #S.3
[Tsourakakis ICDM 2008]

HEP-TH

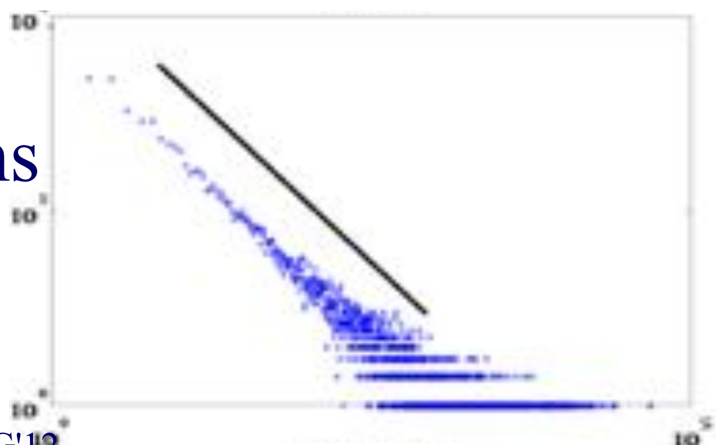


ASN





Epinions



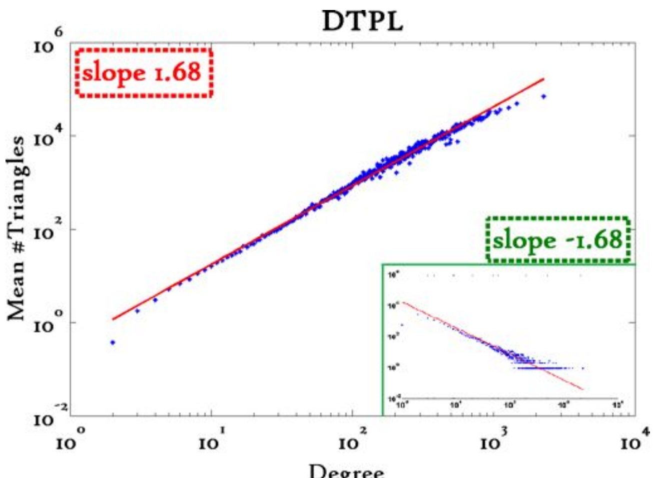
X-axis: # of participating triangles

Y: count (~ pdf)

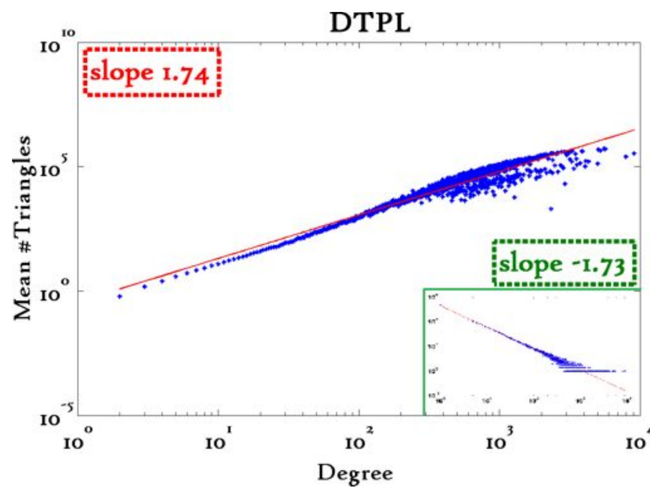
Carnegie Mellon

Triangle Law: #S.4
[Tsourakakis ICDM 2008]

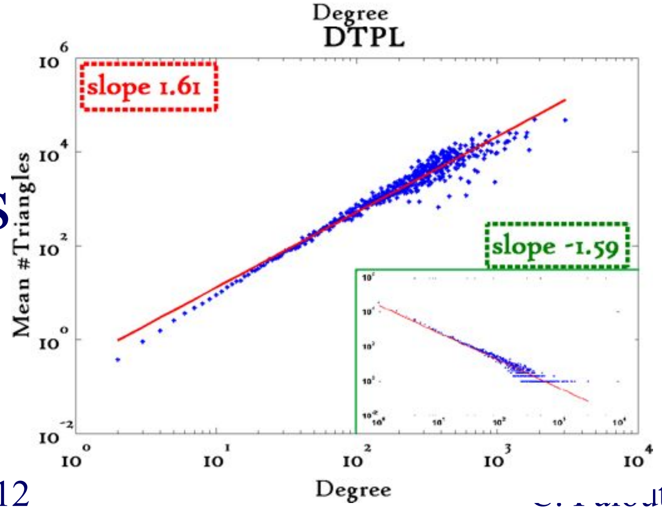
Reuters



SN



Epinions



X-axis: degree

Y-axis: mean # triangles

n friends -> $\sim n^{1.6}$ 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!

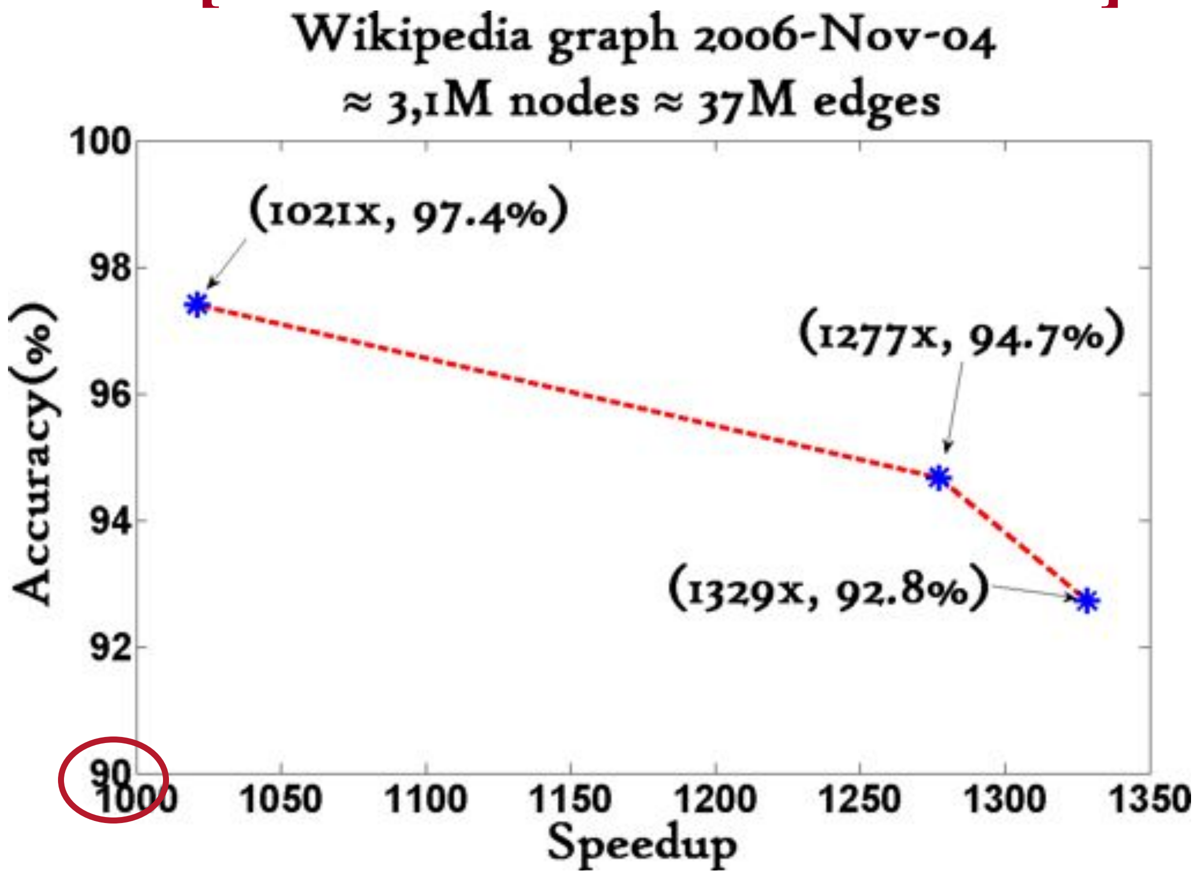
#triangles = 1/6 Sum (λ_i^3)

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

Carnegie Mellon

details

Triangle Law: Computations [Tsourakakis ICDM 2008]



1000x+ speed-up, >90% accuracy

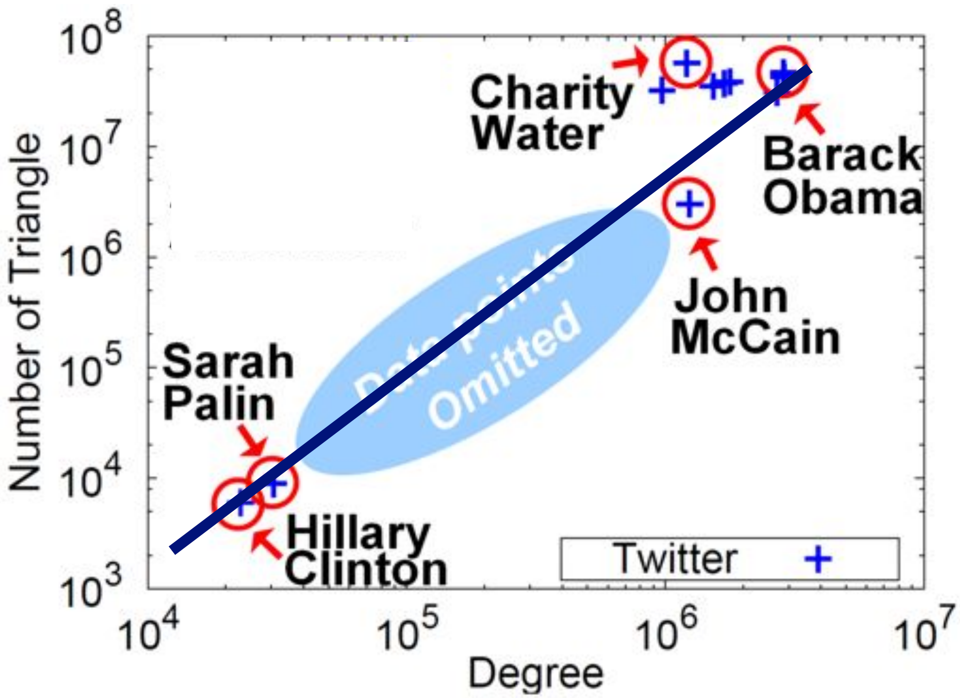
Triangle counting for large graphs?

Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

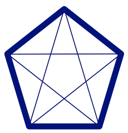
Triangle counting for large graphs?

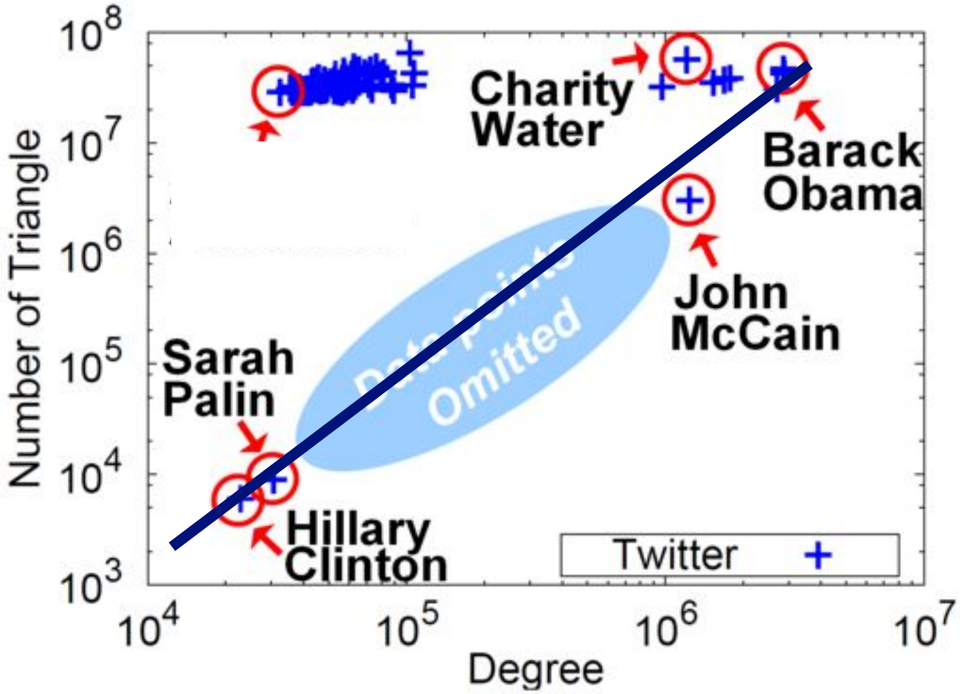


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?

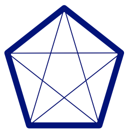


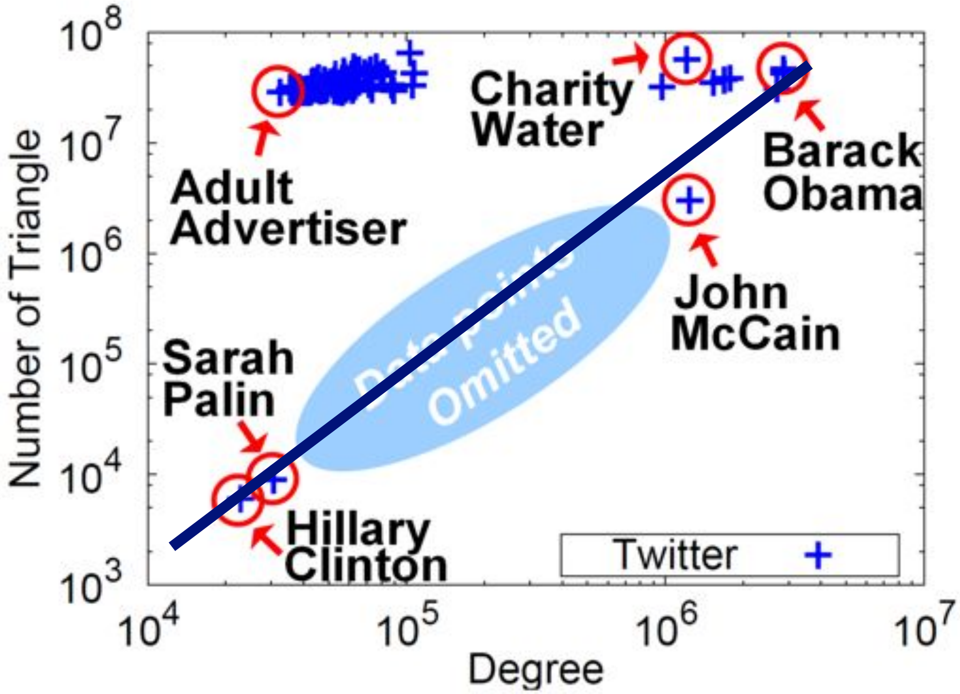


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?





Anomalous nodes in Twitter(~ 3 billion edges)

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

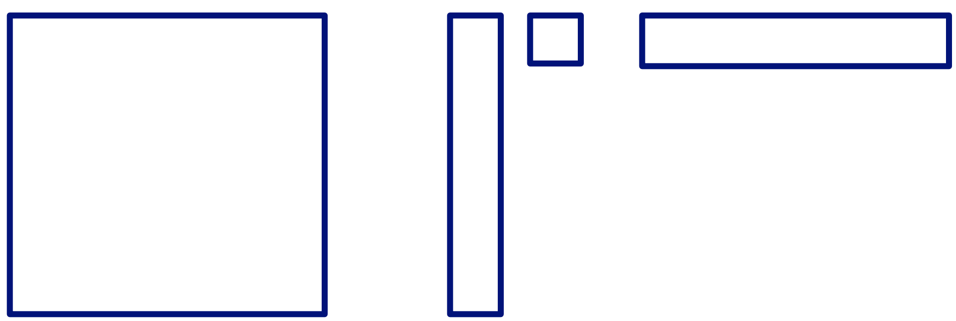
EigenSpokes

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)

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

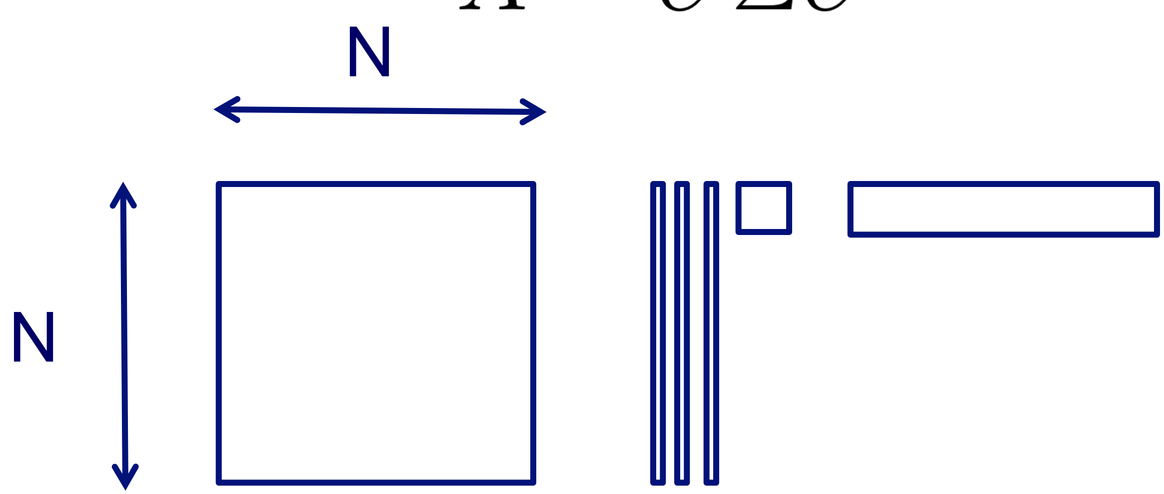


EigenSpokes

details

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

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



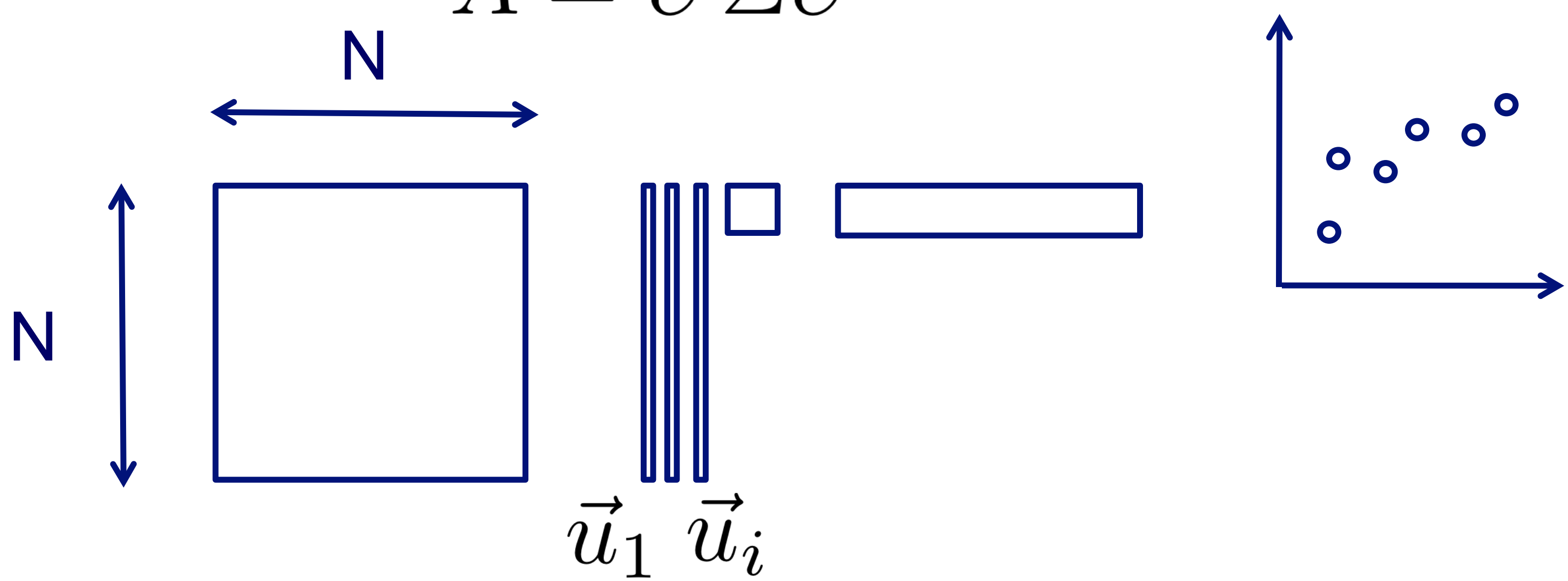
$\vec{u}_1$ $\vec{u}_i$

details

EigenSpokes

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

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

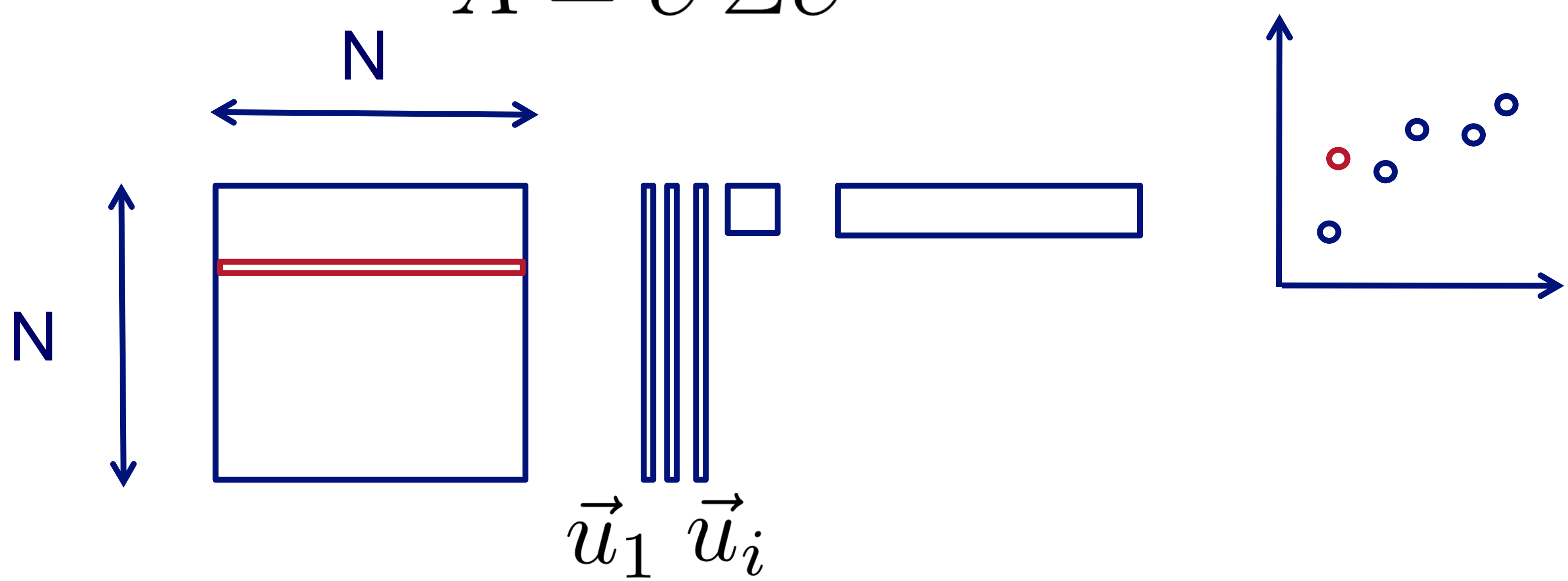


details

EigenSpokes

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

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

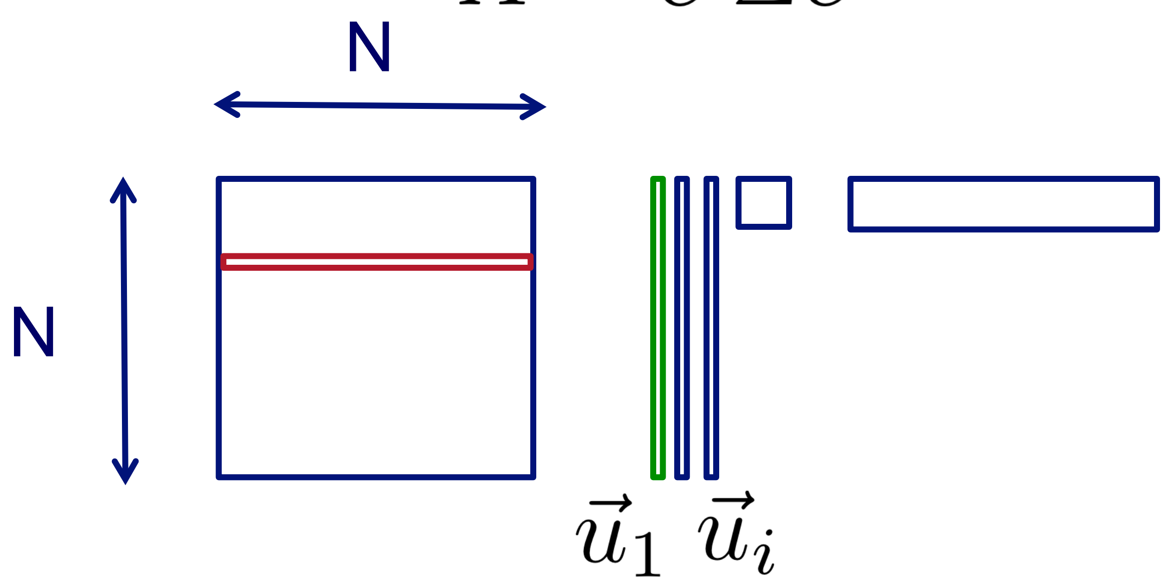


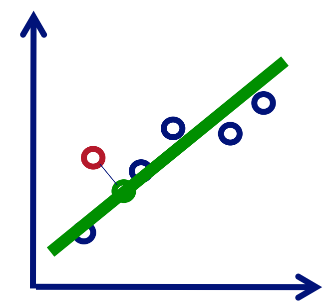
details

EigenSpokes

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

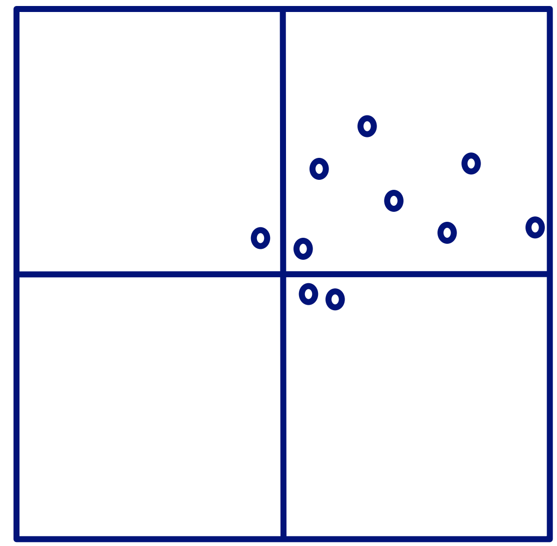
$$A = U \Sigma U^T$$





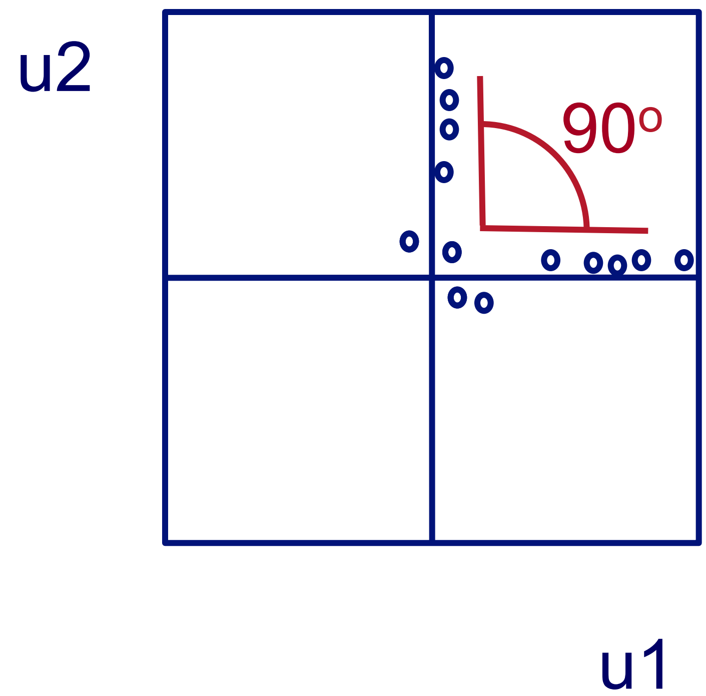
EigenSpokes

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



EigenSpokes

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

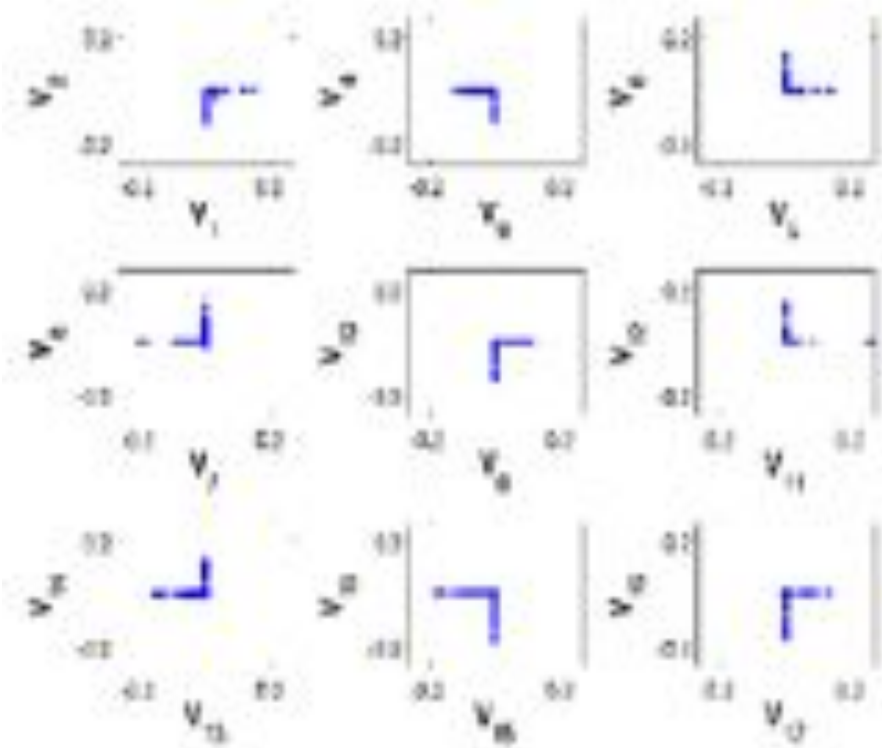


Carnegie Mellon

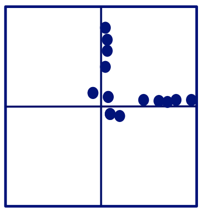
EigenSpokes - pervasiveness

- Present in mobile social graph
  - across time and space

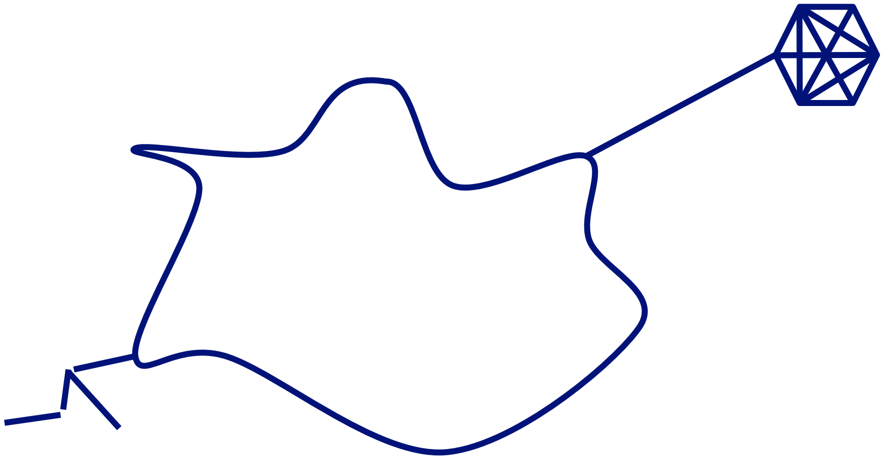
- Patent citation graph



EigenSpokes - explanation

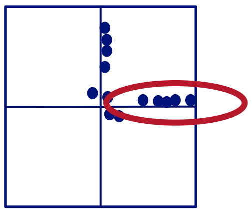


Near-cliques, or near-bipartite-cores, loosely connected

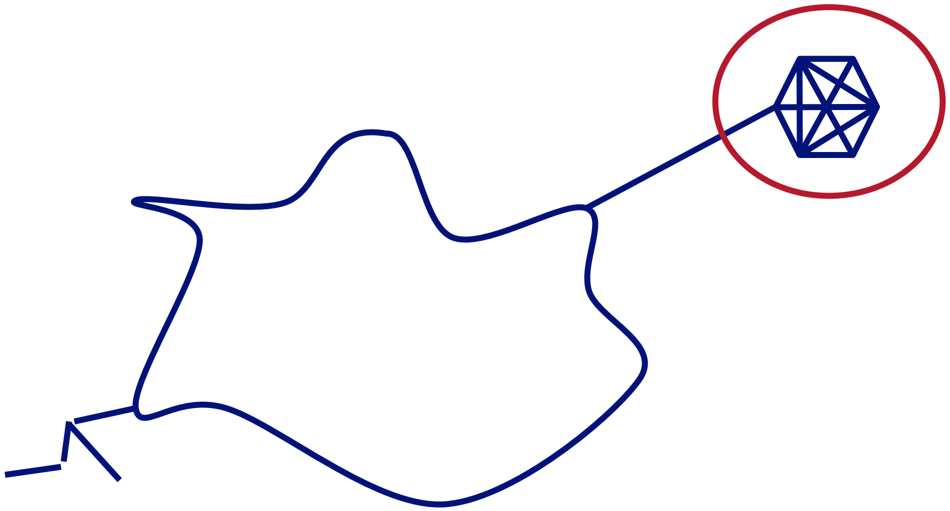


Carnegie Mellon

EigenSpokes - explanation

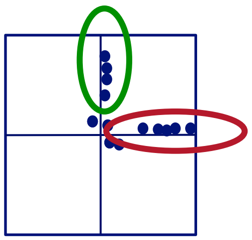


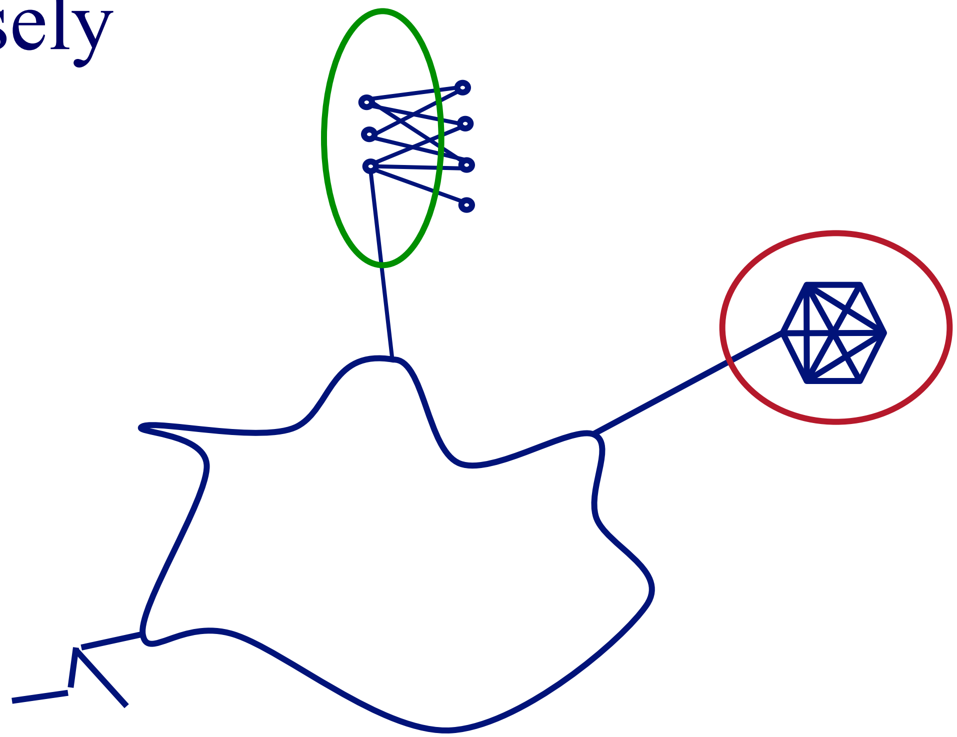
Near-cliques, or near-bipartite-cores, loosely connected



EigenSpokes - explanation

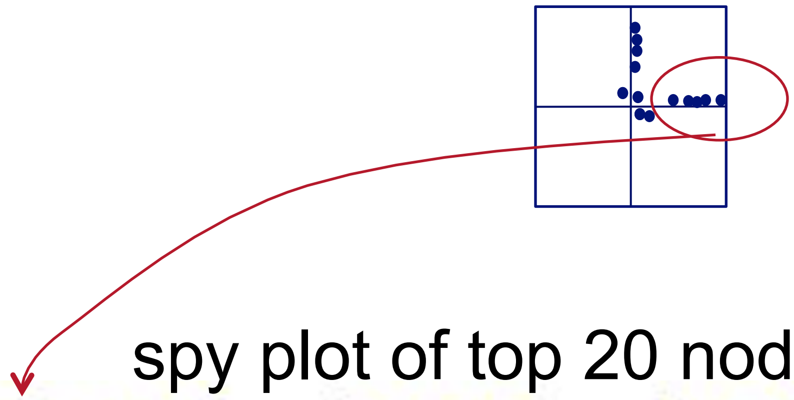
Near-cliques, or near-bipartite-cores, loosely connected



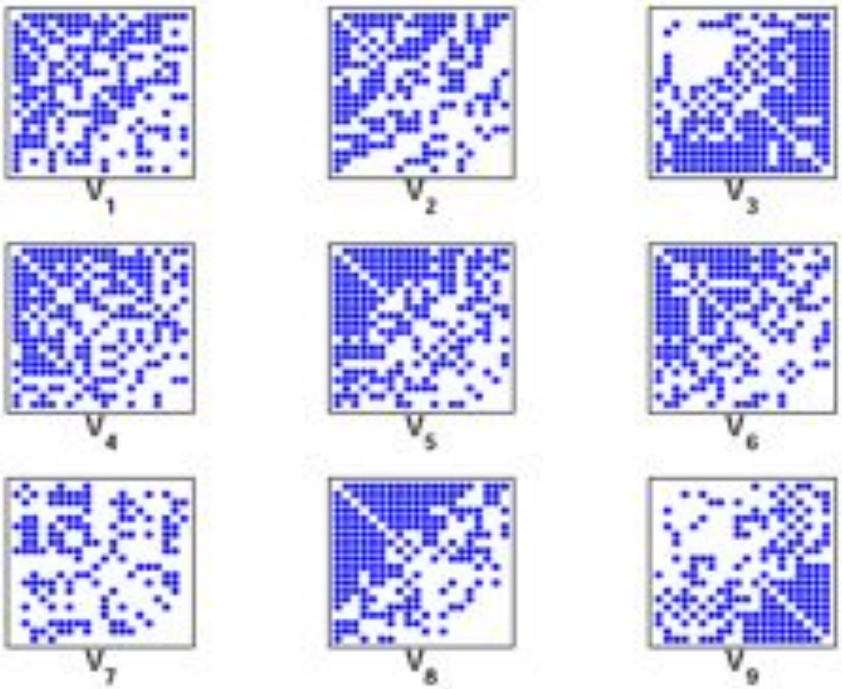


EigenSpokes - explanation

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



spy plot of top 20 nodes



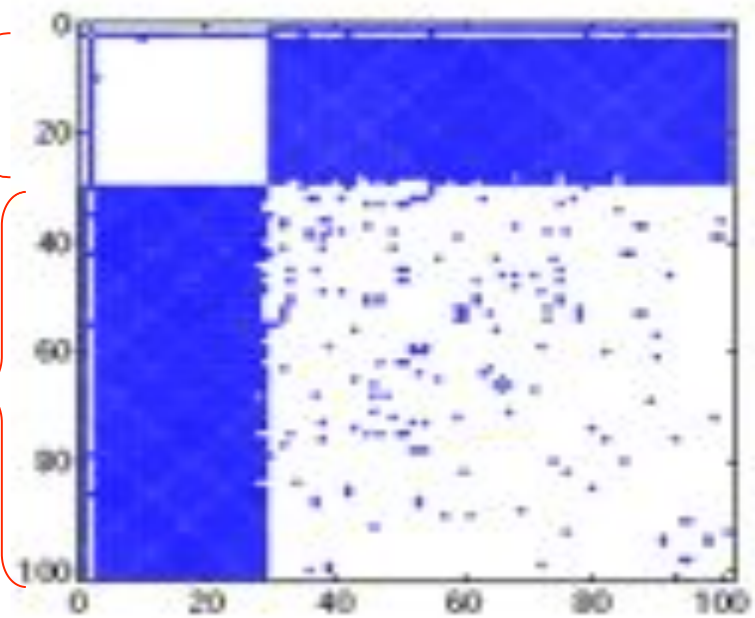
So what?

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

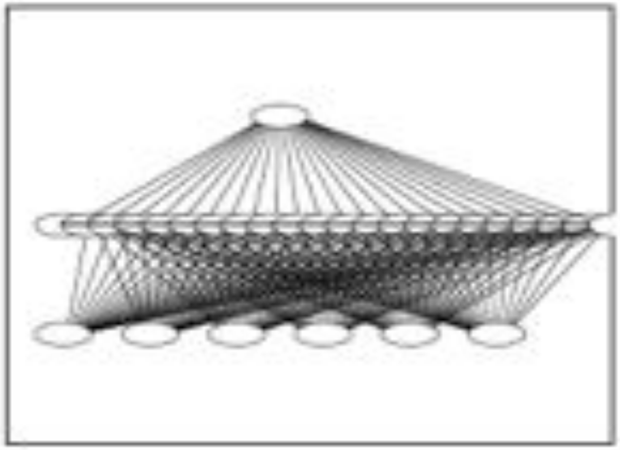
Bipartite Communities!

patents from
same inventor(s)

`cut-and-paste'
bibliography!



magnified bipartite community

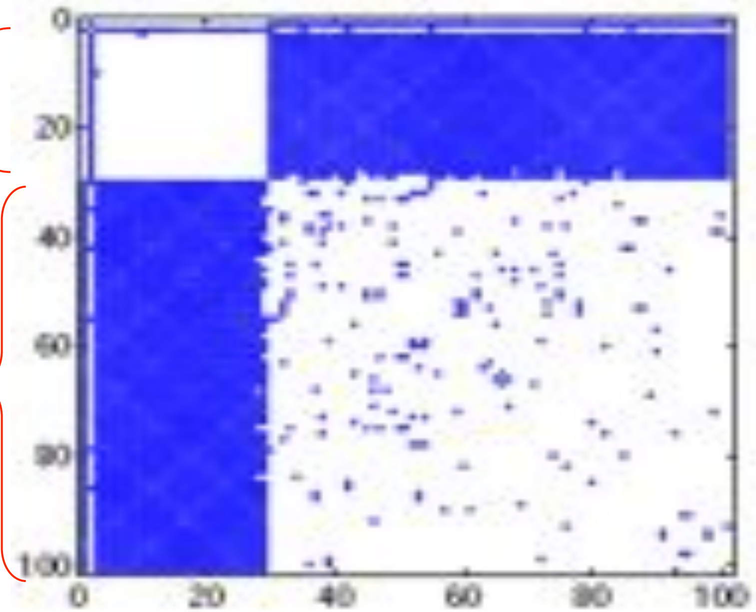


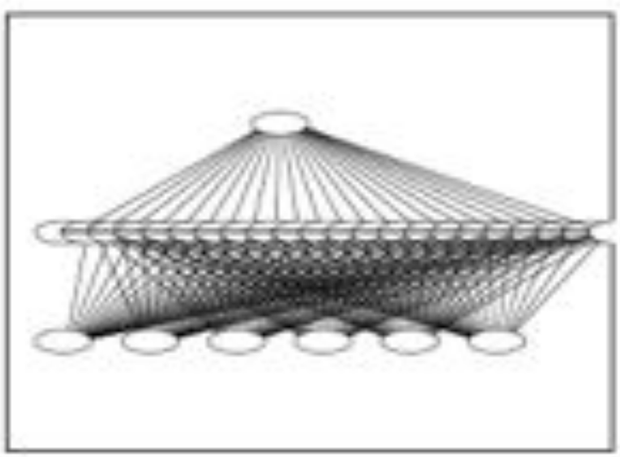
Carnegie Mellon

(maybe, botnets?)

Victim IPs?

Botnet members?





Roadmap

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

Carnegie Mellon

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

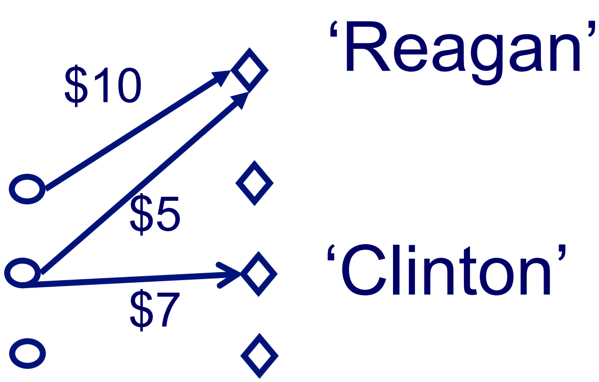
Observation W.1: Fortification

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

Carnegie Mellon

Observation W.1: Fortification

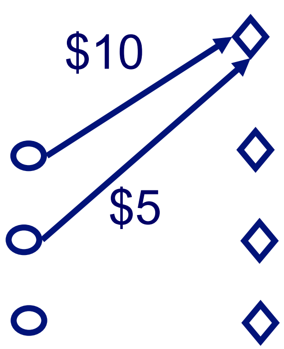
More donors, more $?



Observation W.1: fortification: Snapshot Power Law

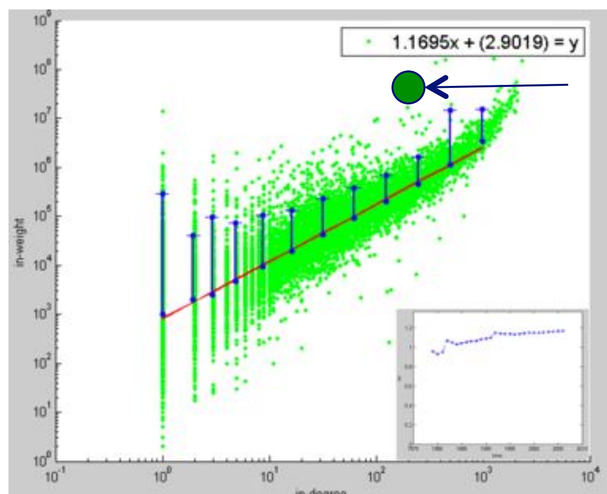
- Weight: super-linear on in-degree
- exponent 'iw': 1.01 < iw < 1.26

More donors, even more $



Orgs-Candidates

In-weights ($)



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

Edges (# donors)

Roadmap

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

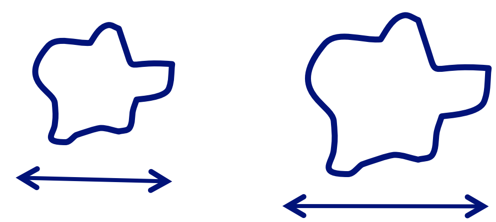
Problem: Time evolution

- with Jure Leskovec (CMU -> Stanford)

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

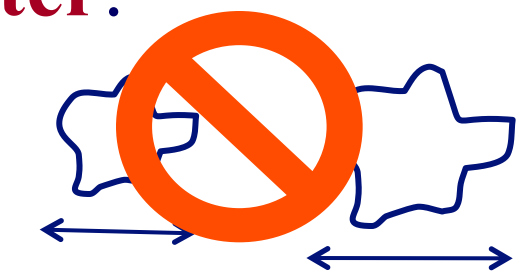
T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?



T.1 Evolution of the Diameter

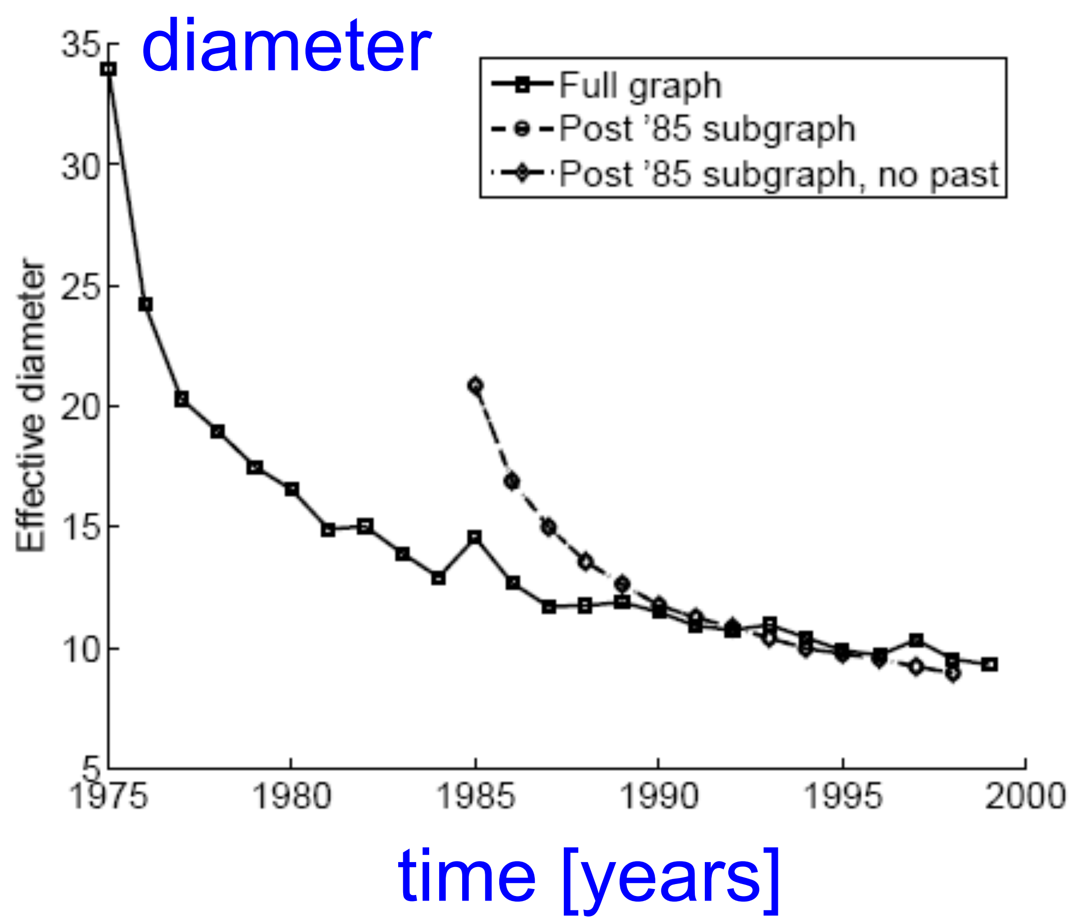
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)



- 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

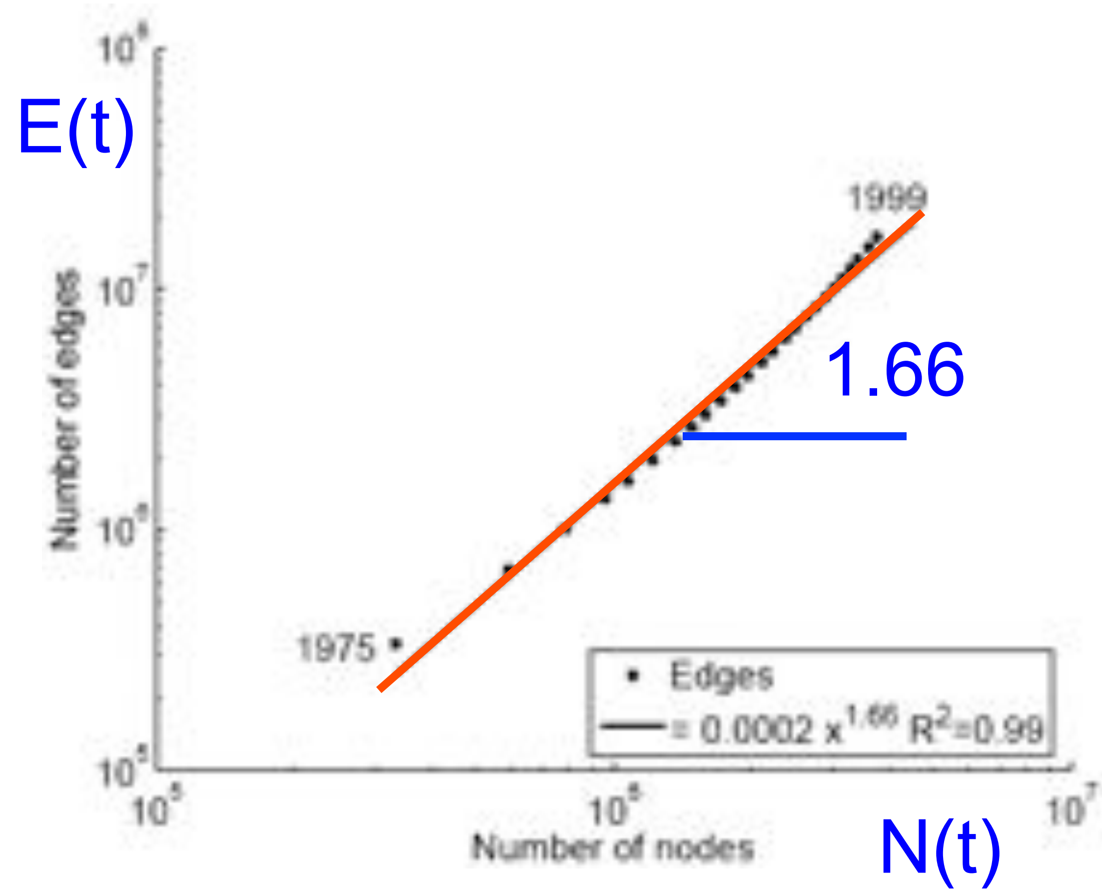
  $$E(t+1) =? 2 * E(t)$$

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

Carnegie Mellon

T.2 Densification – Patent Citations

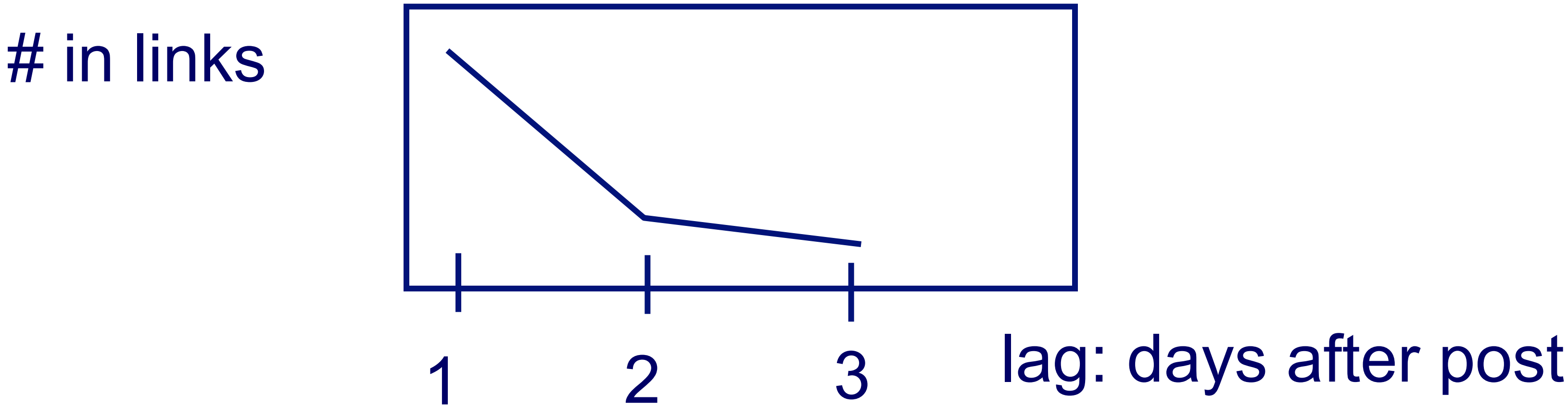
- Citations among patents granted
- @1999
  - 2.9 M nodes
  - 16.5 M edges
- Each year is a datapoint



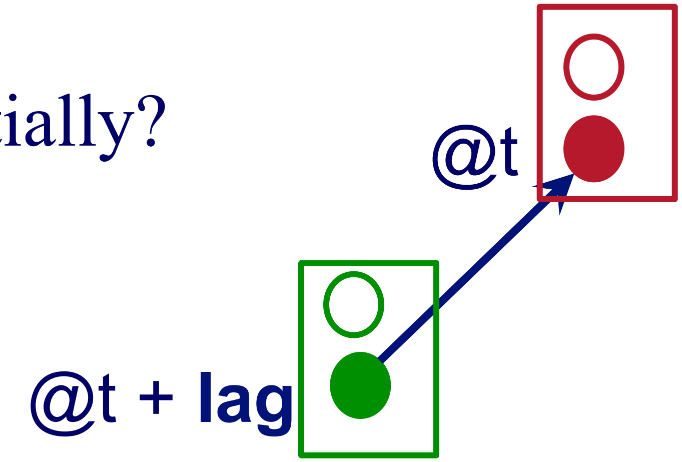
Roadmap

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

T.3 : popularity over time

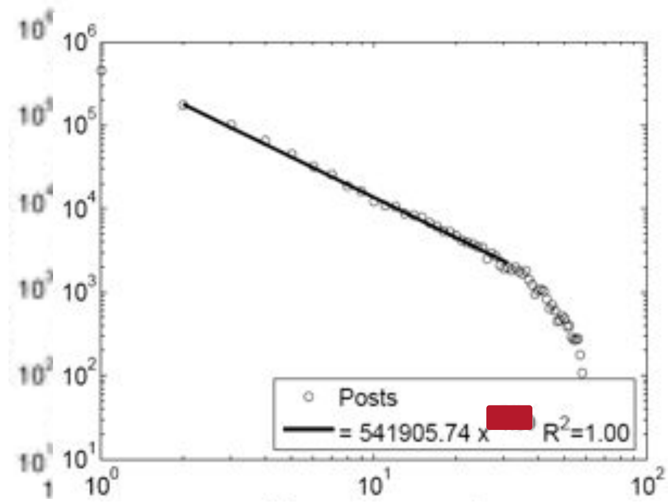


Post popularity drops-off – exponentially?



T.3 : popularity over time

in links (**log**)



days after post (**log**)

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

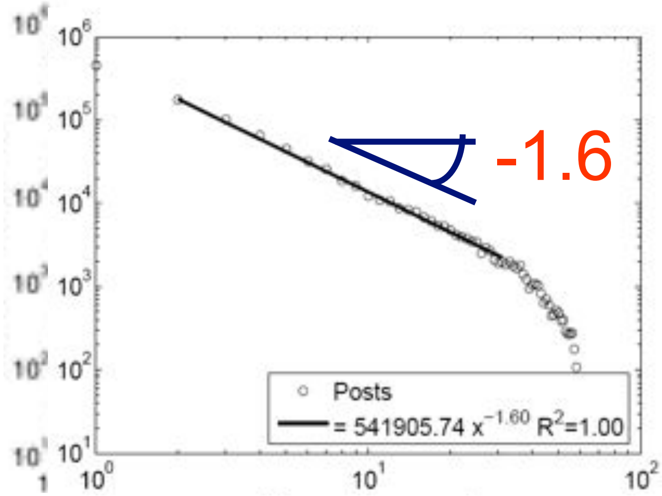
POWER LAW!

Exponent?

Carnegie Mellon

T.3 : popularity over time

in links (**log**)



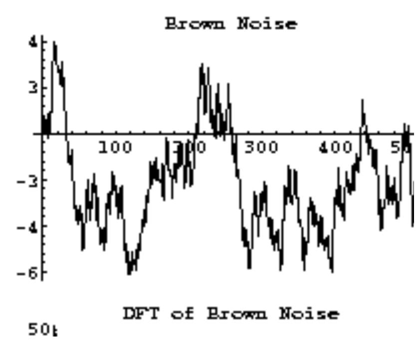
days after post (**log**)

Post popularity drops-off – exponentially?

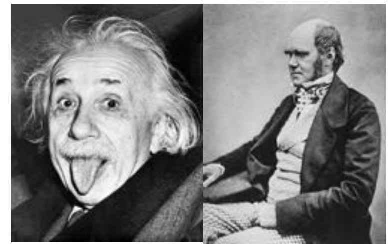
POWER LAW!

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]

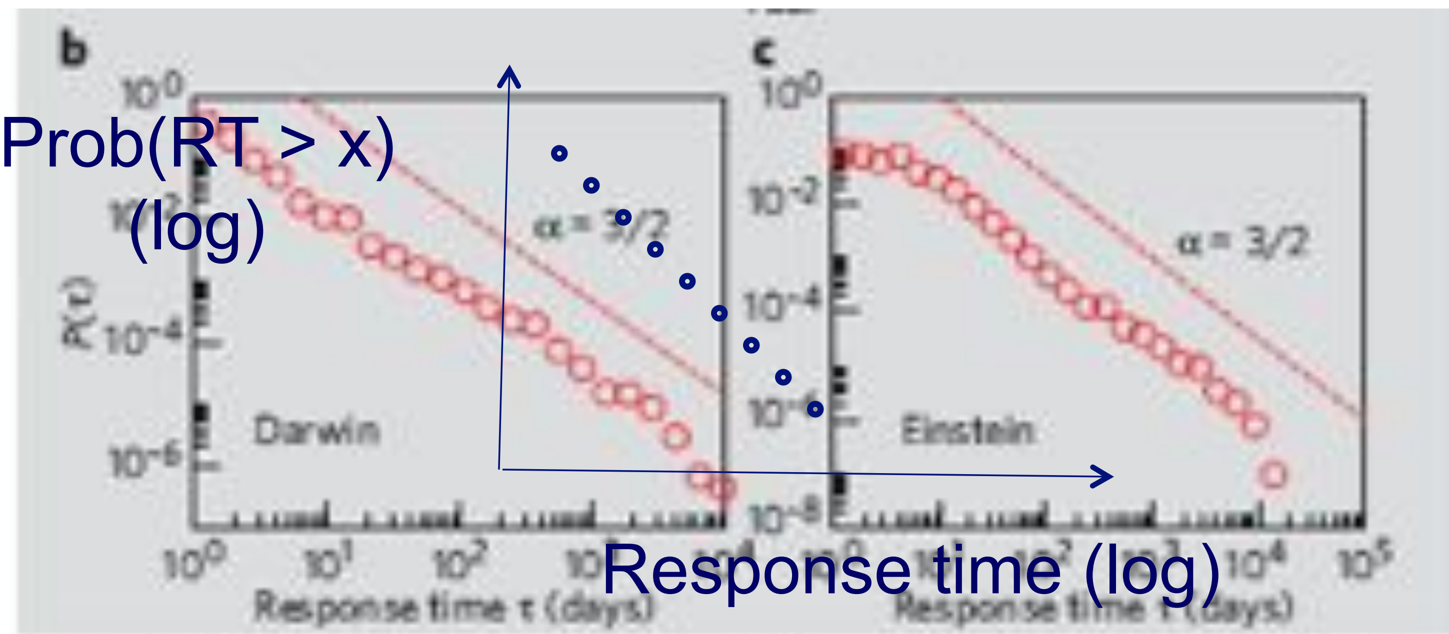


Figure 1 | The correspondence patterns of Darwin and Einstein.

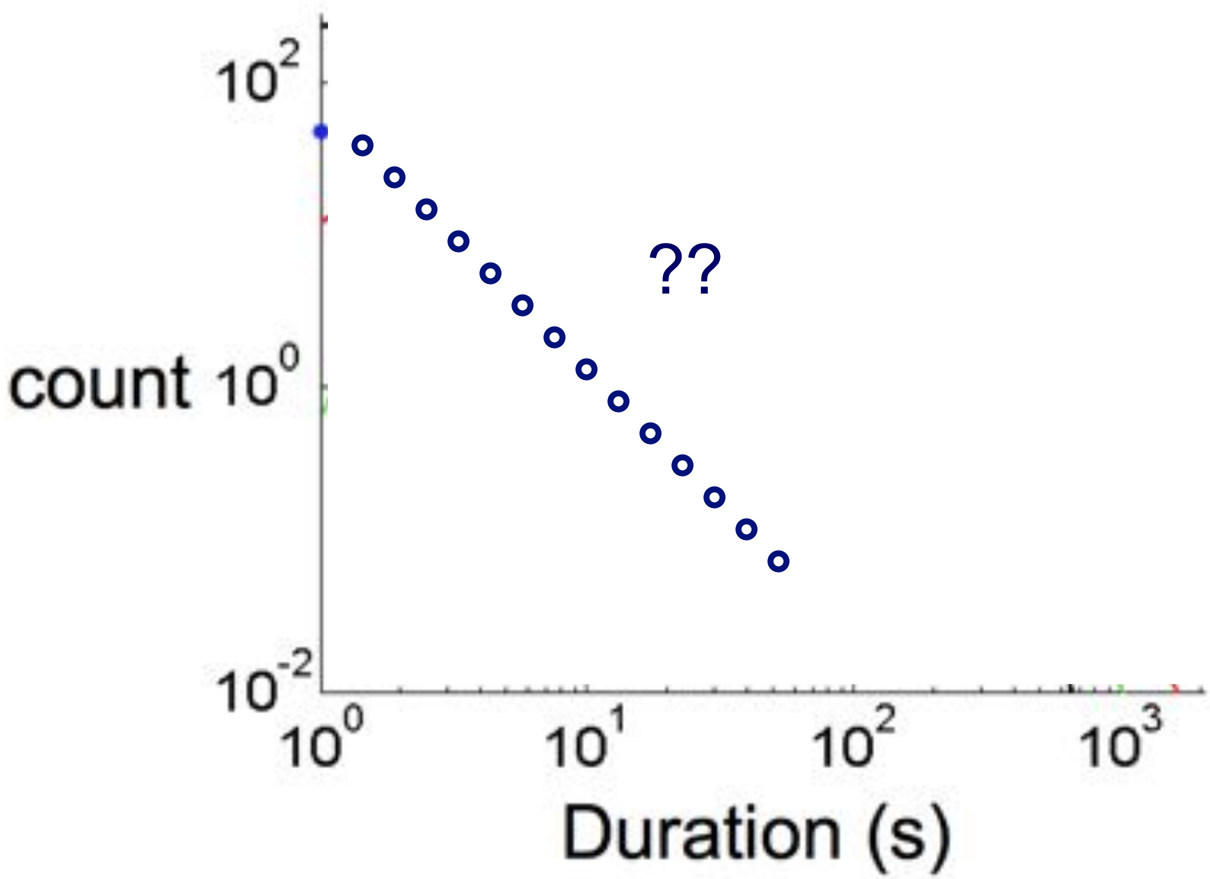
T.4: duration of phonecalls

Surprising Patterns for the Call Duration Distribution of Mobile Phone Users

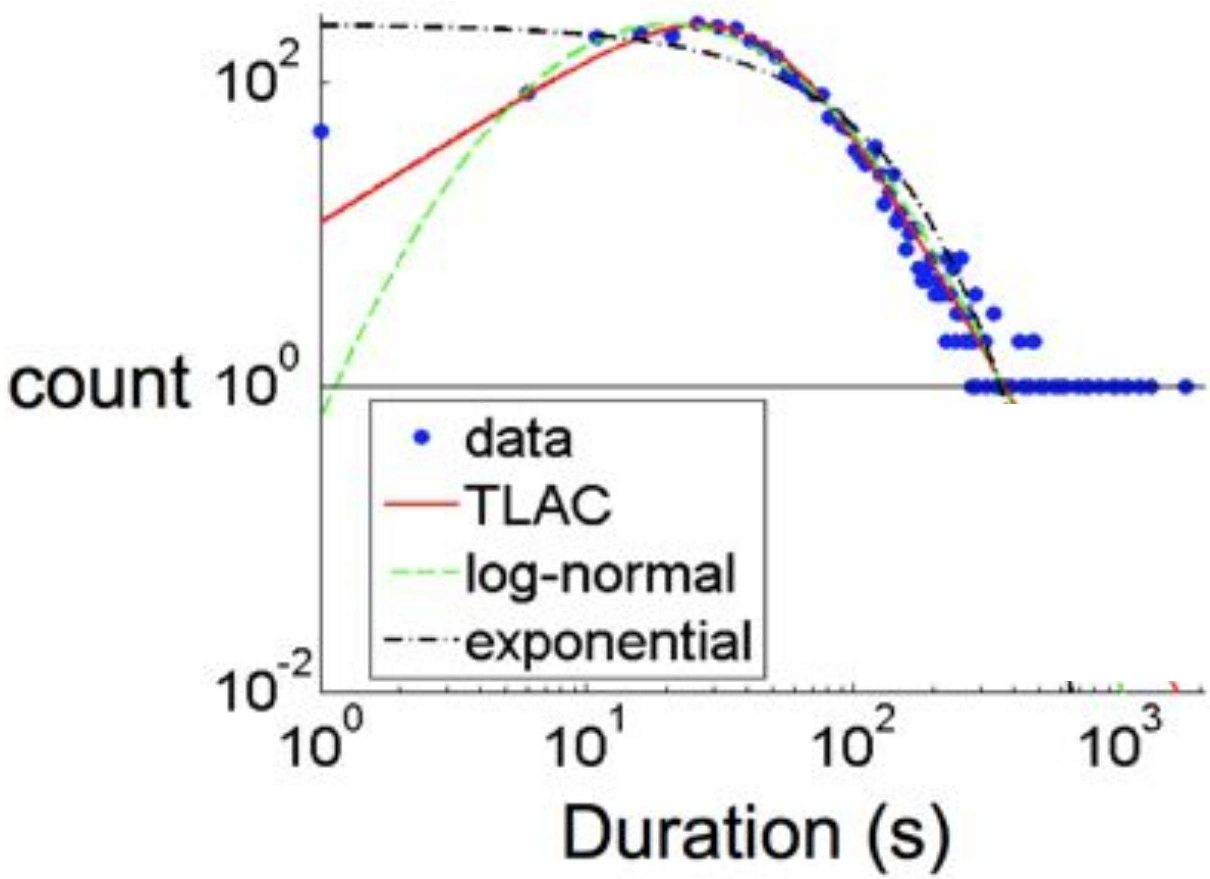
Pedro O. S. Vaz de Melo, Leman Akoglu, Christos Faloutsos, Antonio A. F. Loureiro

PKDD 2010

Probably, power law (?)



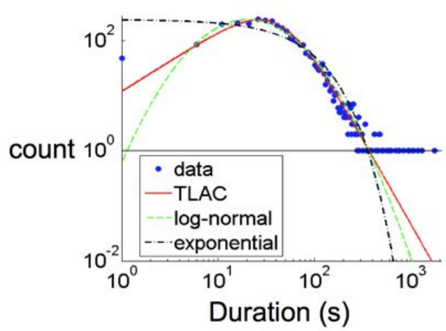
No Power Law!



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

'TLaC: Lazy Contractor'

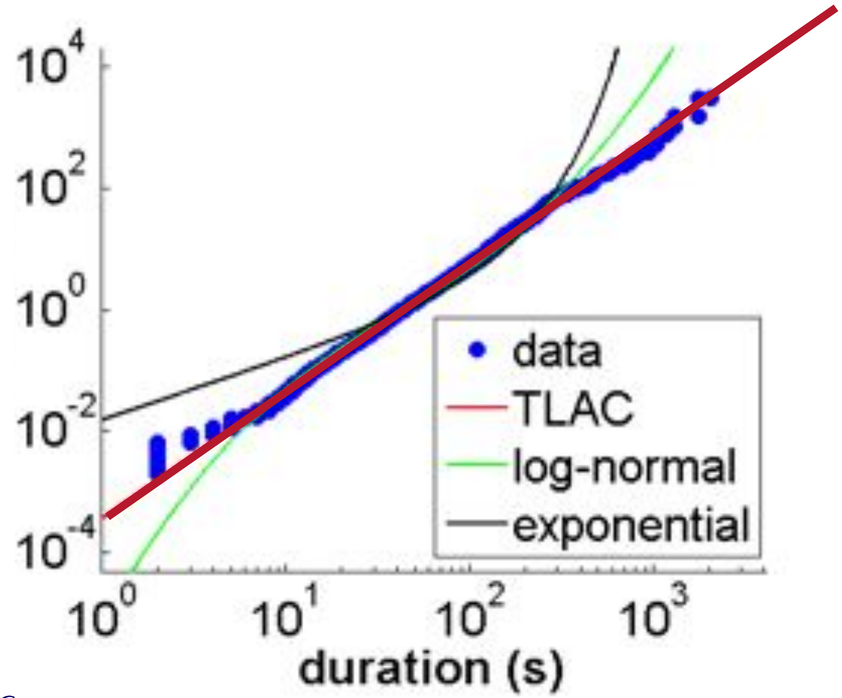
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

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

== power law

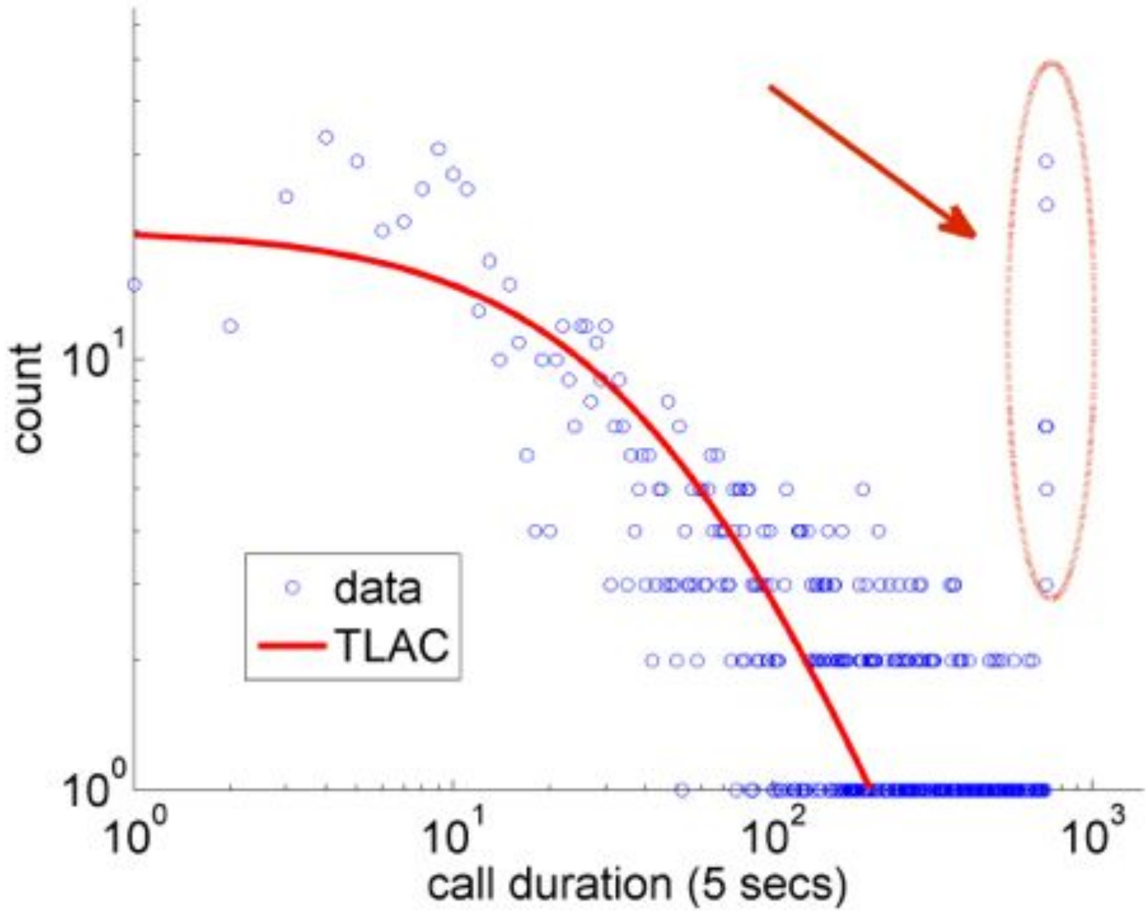


Data Description

- 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)

Carnegie Mellon

Outliers:



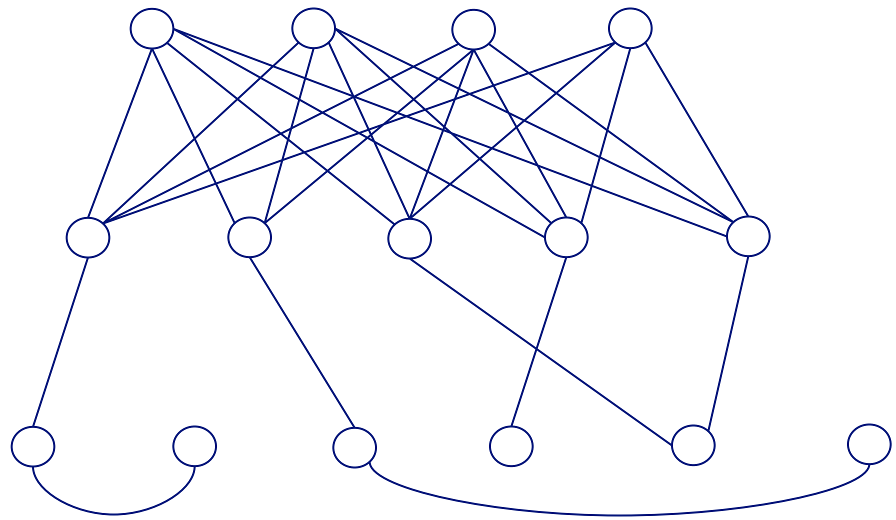
Carnegie Mellon

Roadmap

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief Propagation
  - Tensors
  - Spike analysis
- Problem#3: Scalability
- Conclusions

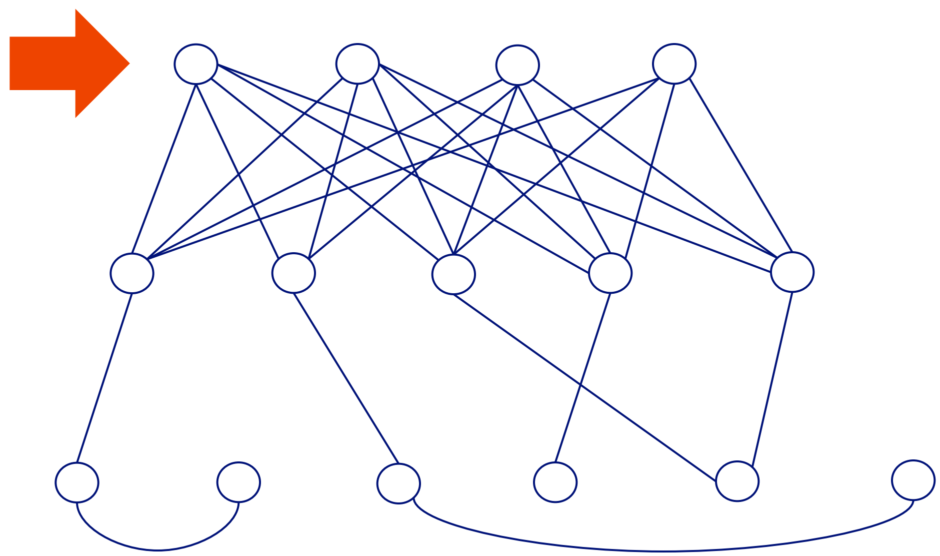
E-bay Fraud detection

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

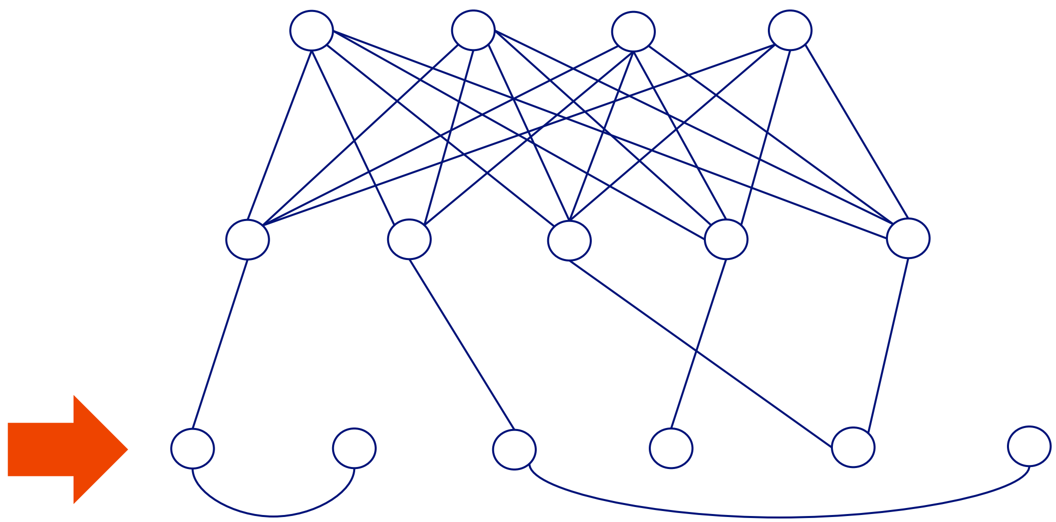


Carnegie Mellon

E-bay Fraud detection

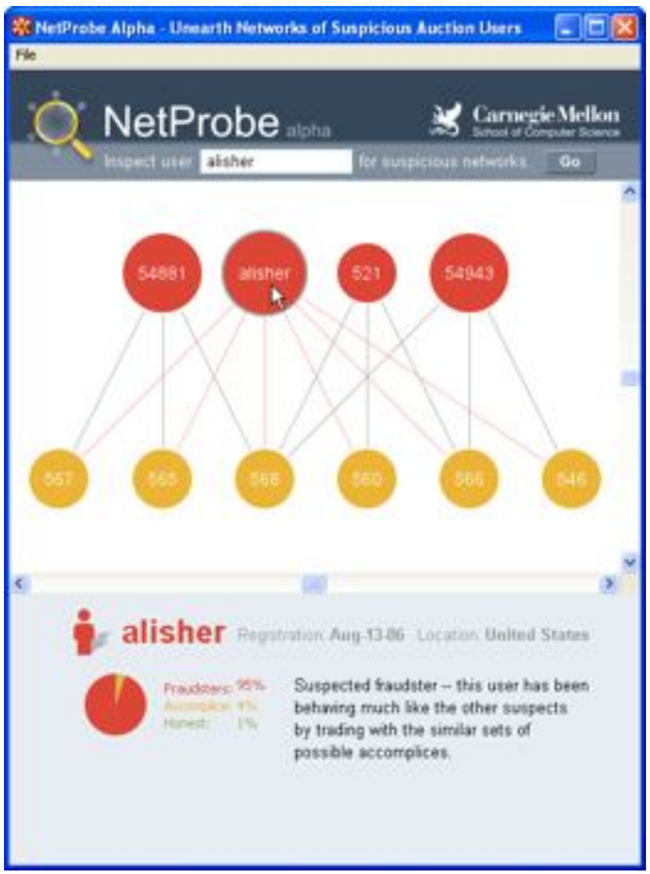


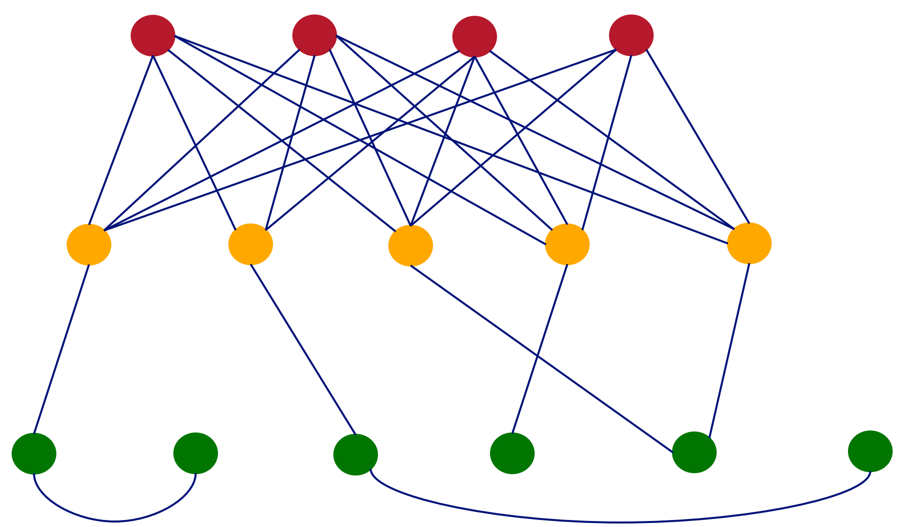
E-bay Fraud detection



Carnegie Mellon

E-bay Fraud detection - NetProbe





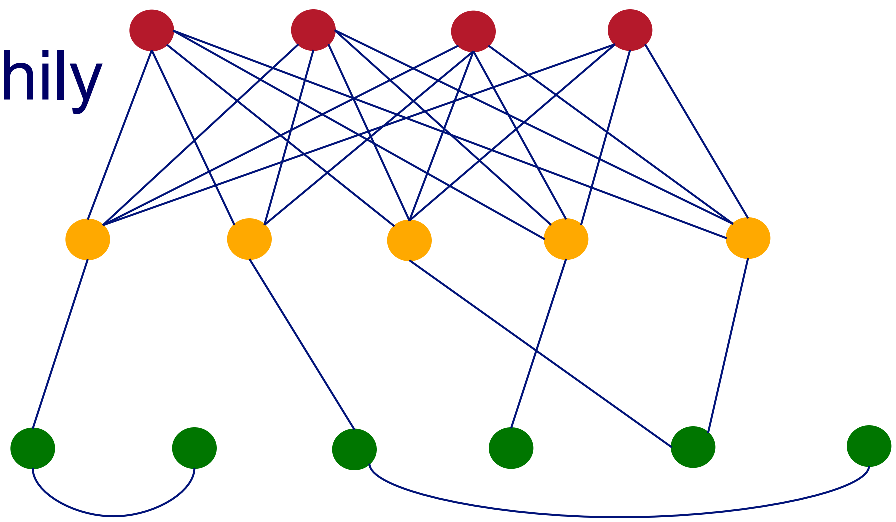
Carnegie Mellon

E-bay Fraud detection - NetProbe

Compatibility matrix

	F	A	H
F		99%	
A	99%		
H		49%	49%

heterophily



Popular press

And less desirable attention:

- E-mail from ‘Belgium police’ (‘copy of your code?’)

Roadmap

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief Propagation
  - Tensors
  - Spike analysis
- Problem#3: Scalability
- Conclusions

GigaTensor: Scaling Tensor Analysis Up By 100 Times – Algorithms and Discoveries

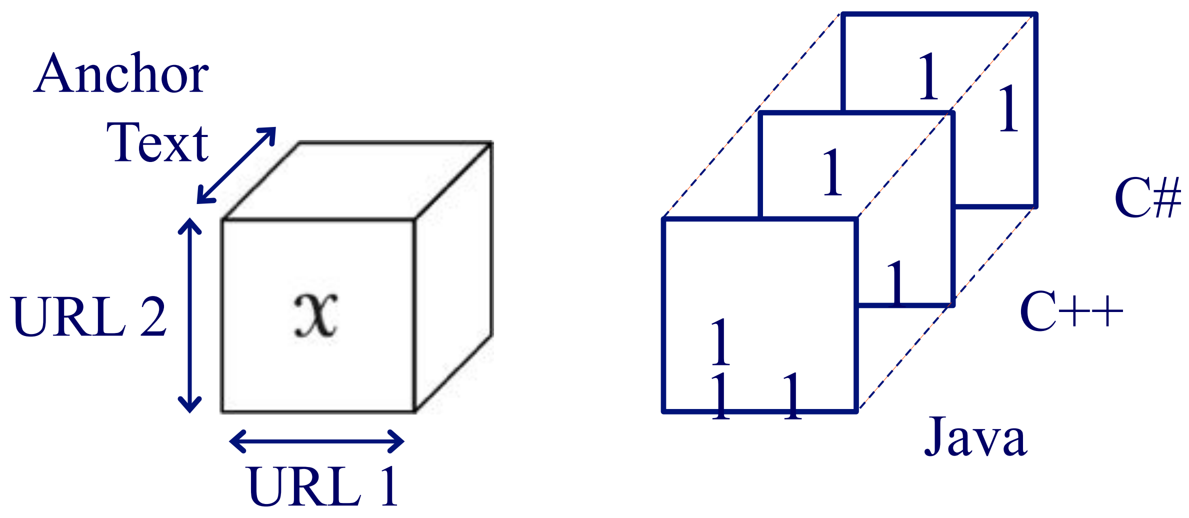
U Kang	Evangelos Papalexakis	Abhay Harpale	Christos Faloutsos

KDD'12

Carnegie Mellon

Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Hyperlinks &anchor text [Kolda+,05]

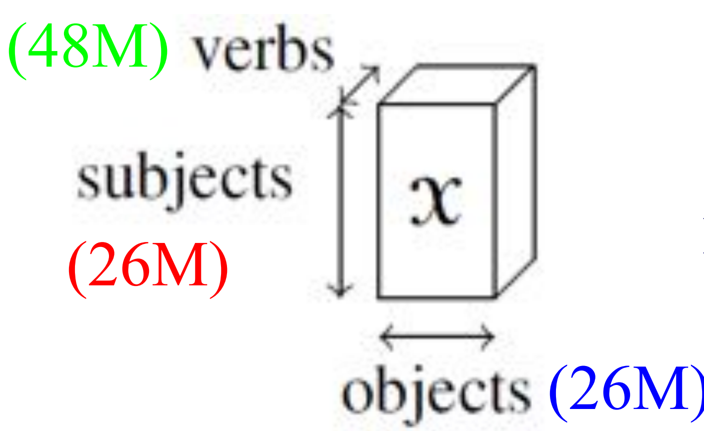


Background: Tensor

- Tensors (=multi-dimensional arrays) are everywhere
  - Sensor stream (time, location, type)
  - Predicates (subject, verb, object) in knowledge base

"Eric Clapton plays guitar"

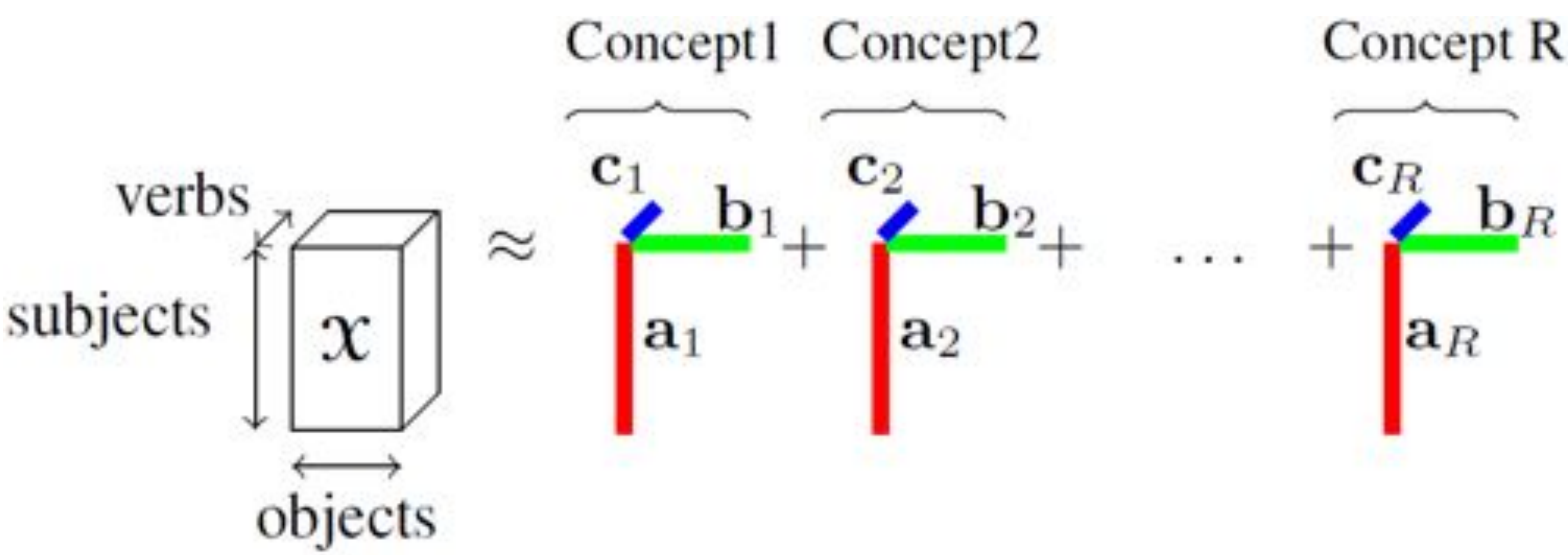
"Barrack Obama is the president of U.S."



NELL (Never Ending Language Learner) data
Nonzeros =144M

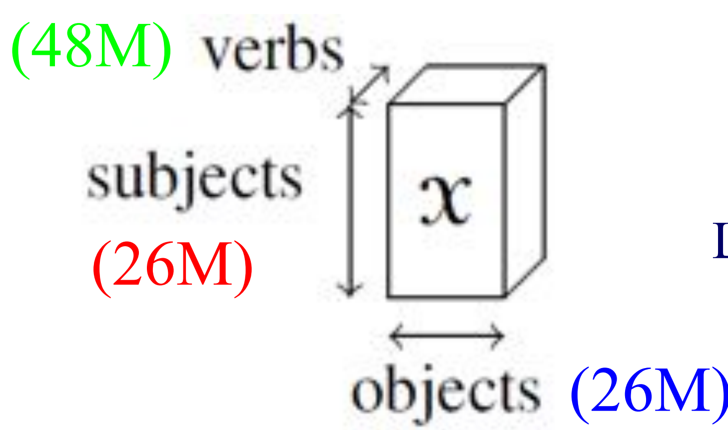
Problem Definition

- How to decompose a billion-scale tensor?
  - Corresponds to SVD in 2D case



Problem Definition

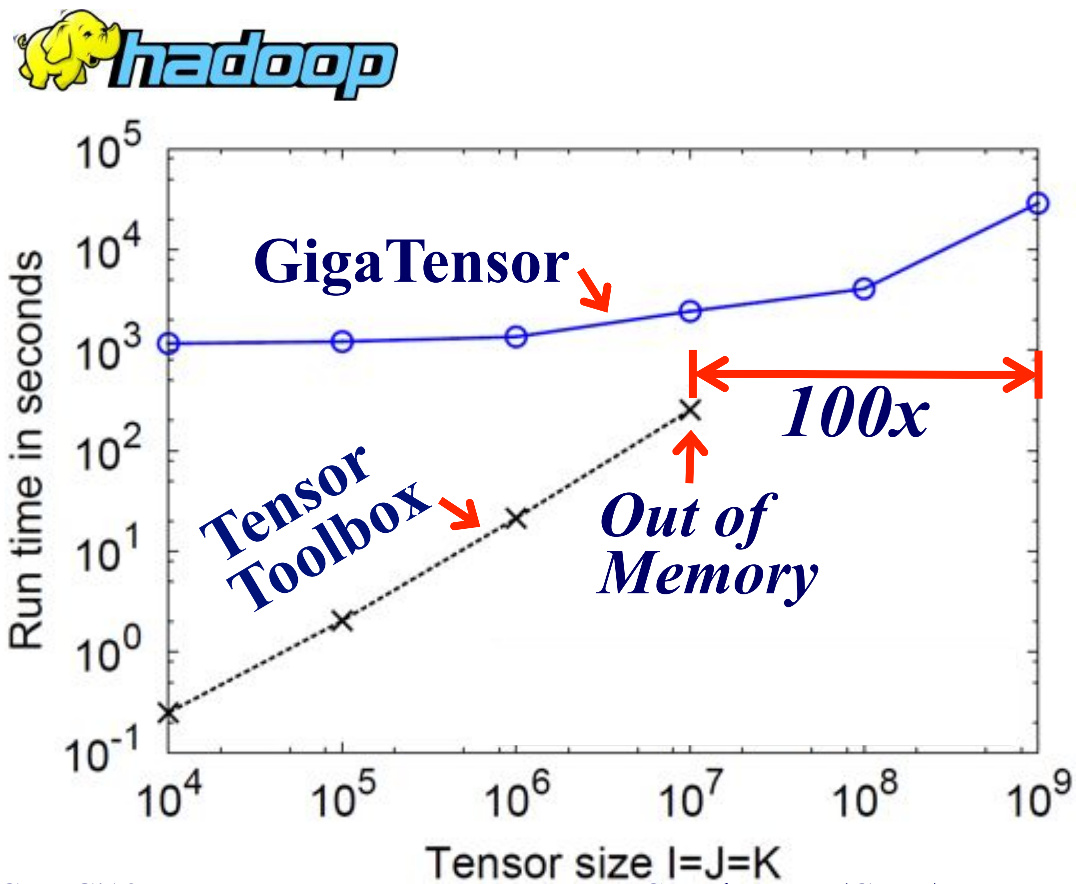
- Q1: Dominant concepts/topics?
- Q2: Find synonyms to a given noun phrase?
- (and how to scale up: |data| > RAM)

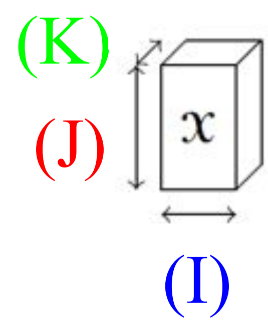


NELL (Never Ending Language Learner) data
Nonzeros =144M

Experiments

- GigaTensor solves ***100x*** larger problem



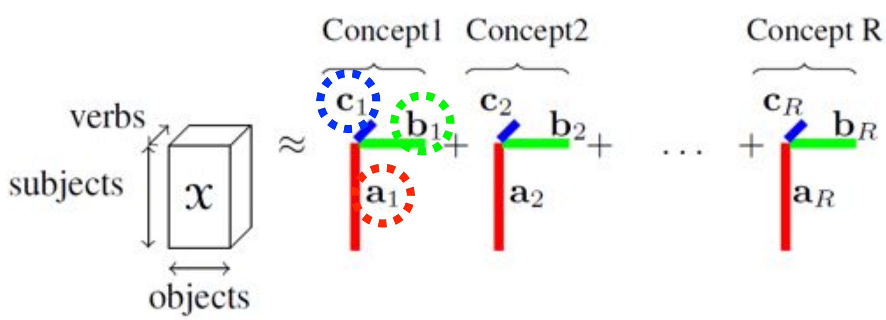


Number of nonzero = I / 50

Carnegie Mellon

A1: Concept Discovery

- Concept Discovery in Knowledge Base



Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	‘np1’ ‘stream’ ‘np2’
file	software	‘np1’ ‘marketing’ ‘np2’
data	suite	‘np1’ ‘dating’ ‘np2’
Concept 2: "Credit Cards"		
credit	information	‘np1’ ‘card’ ‘np2’
Credit	debt	‘np1’ ‘report’ ‘np2’
library	number	‘np1’ ‘cards’ ‘np2’
Concept 3: "Health System"		
health	provider	‘np1’ ‘care’ ‘np2’
child	providers	’np’ ‘insurance’ ‘np2’
home	system	‘np1’ ‘service’ ‘np2’
Concept 4: "Family Life"		
life	rest	‘np2’ ‘of’ ‘my’ ‘np1’
family	part	‘np2’ ‘of’ ‘his’ ‘np1”
body	years	‘np2’ ‘of’ ‘her’ ‘np1’

Carnegie Mellon

A1: Concept Discovery

Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	'np1' 'stream' 'np2'
file	software	'np1' 'marketing' 'np2'
data	suite	'np1' 'dating' 'np2'
Concept 2: "Credit Cards"		
credit	information	'np1' 'card' 'np2'
Credit	debt	'np1' 'report' 'np2'
library	number	'np1' 'cards' 'np2'
Concept 3: "Health System"		
health	provider	'np1' 'care' 'np2'
child	providers	'np' 'insurance' 'np2'
home	system	'np1' 'service' 'np2'

A2: Synonym Discovery

(Given) Noun Phrase	(Discovered) Potential Synonyms
pollutants	dioxin, sulfur dioxide, greenhouse gases, particulates, nitrogen oxide, air pollutants, cholesterol
disabilities	infections, dizziness, injuries, diseases, drowsiness, stiffness, injuries
vodafone	verizon, comcast
Christian history	European history, American history, Islamic history, history
disbelief	dismay, disgust, astonishment

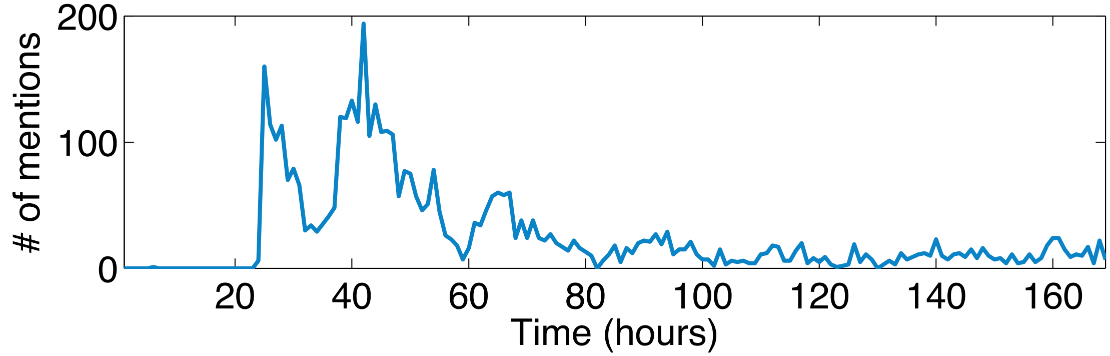
Roadmap

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief propagation
  - Tensors
  - Spike analysis
- Problem#3: Scalability -PEGASUS
- Conclusions

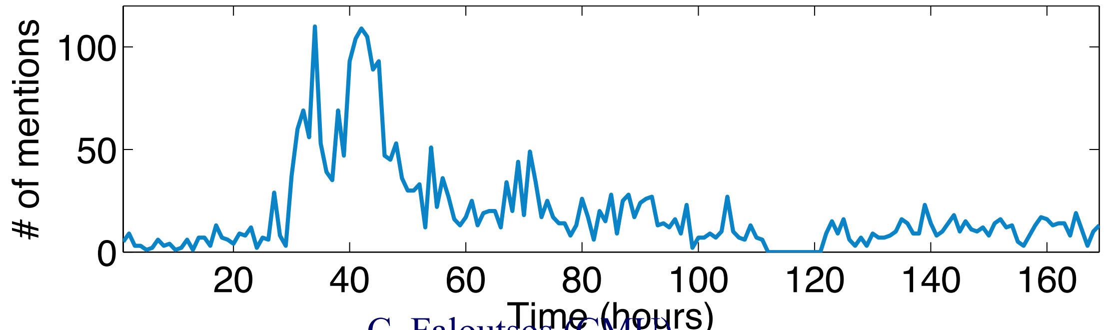
Rise and fall patterns in social media

- Meme (# of mentions in blogs)
  - short phrases Sourced from U.S. politics in 2008

"you can put lipstick on a pig"



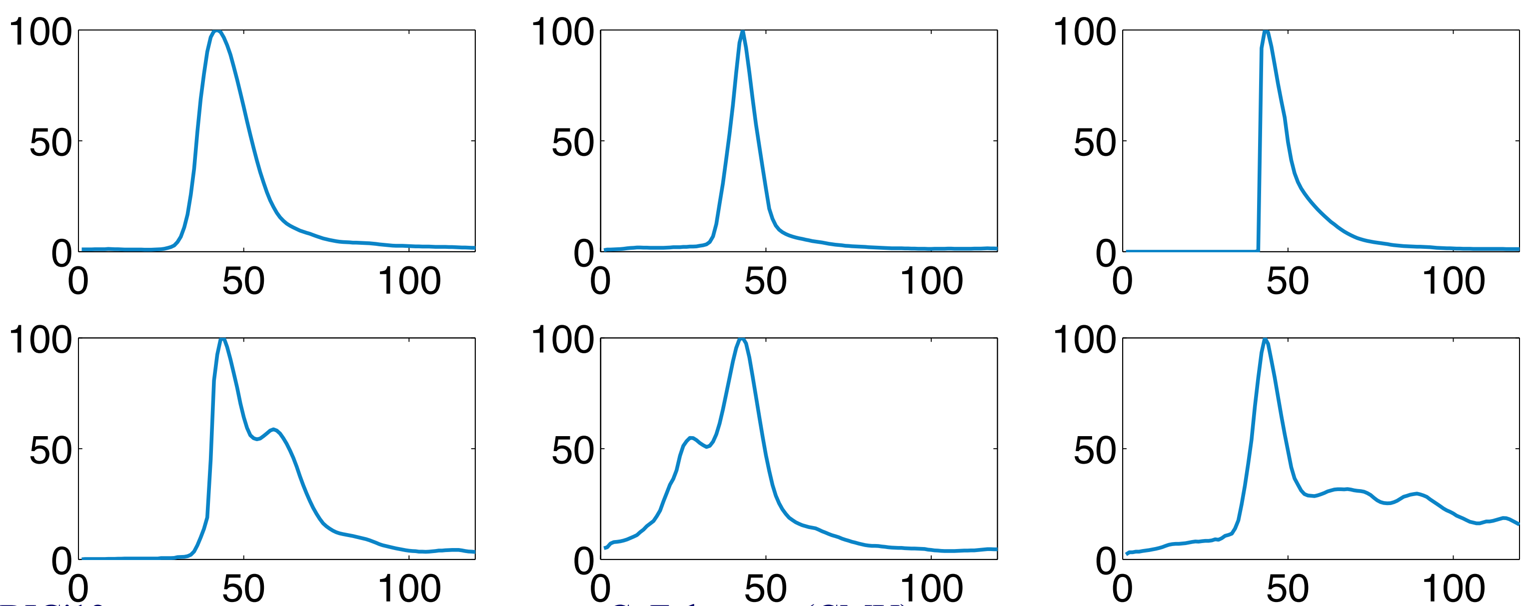
"yes we can"



Carnegie Mellon

Rise and fall patterns in social media

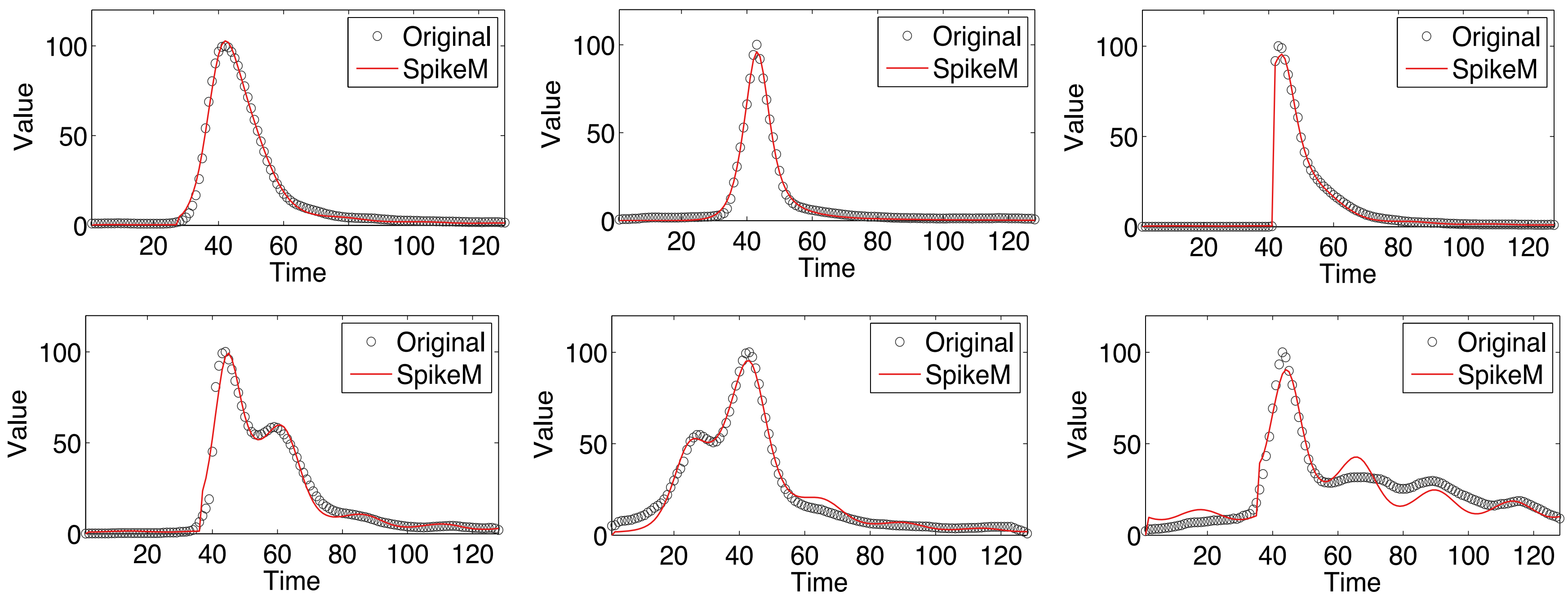
- Can we find a unifying model, which includes these patterns?
  - **four** classes on YouTube [Crane et al. ’08]
  - **six** classes on Meme [Yang et al. ’11]



Carnegie Mellon

Rise and fall patterns in social media

- Answer: YES!

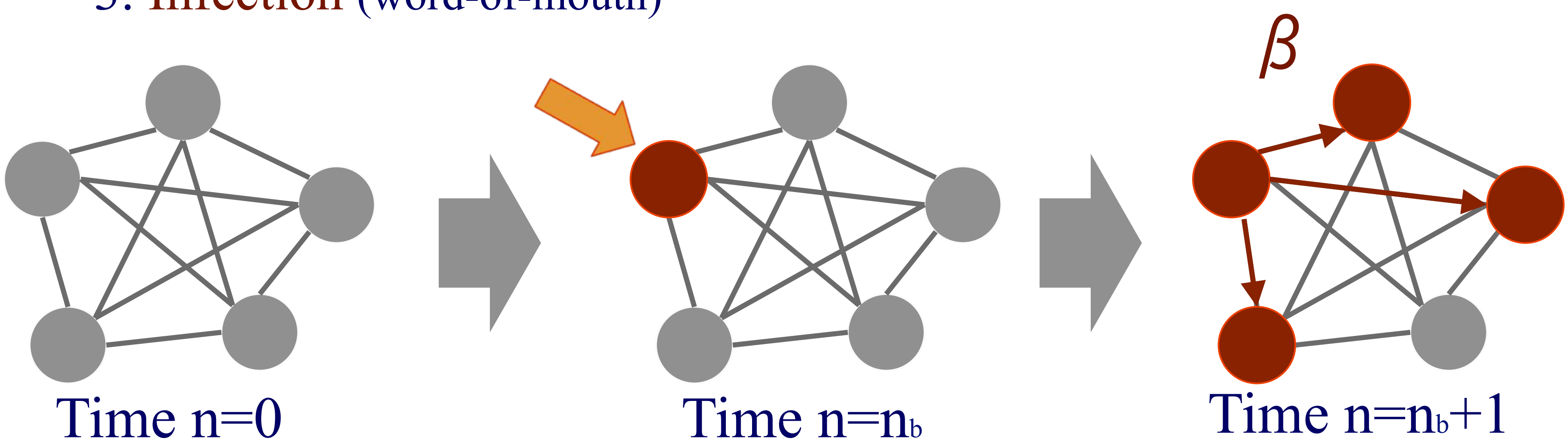


- We can represent **all patterns** by **single model**

In Matsubara+ SIGKDD 2012

Main idea - SpikeM

- 1. Un-informed bloggers (uninformed about rumor)
- 2. External shock at time n_b (e.g, breaking news)
- 3. Infection (word-of-mouth)



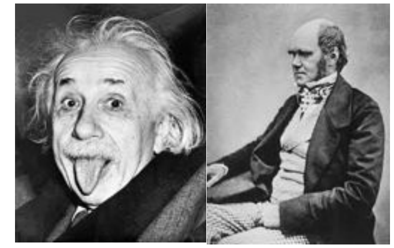
Infectiveness of a blog-post at age n:

β – Strength of infection (quality of news)

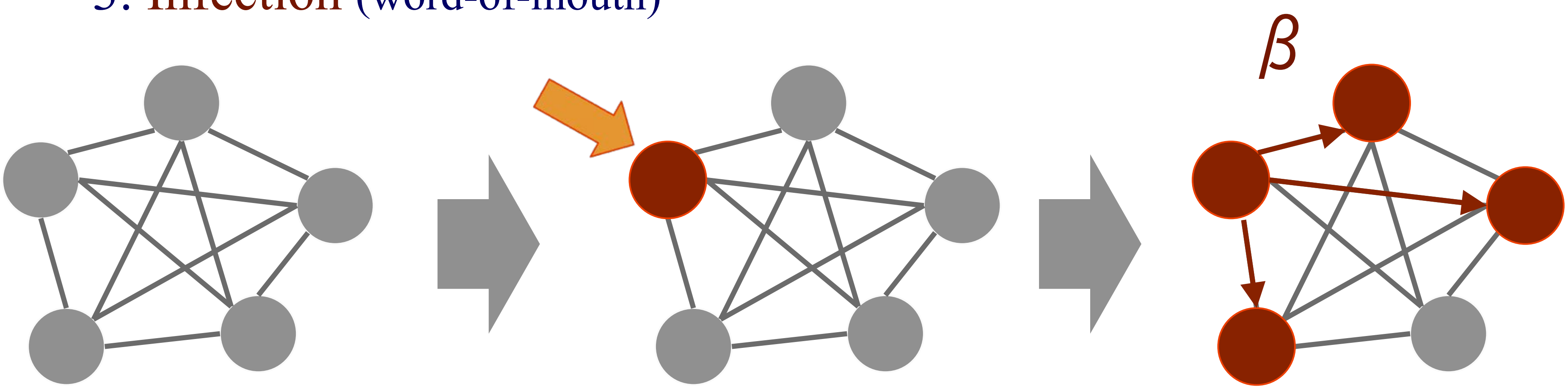
$f(n)$ – Decay function

Carnegie Mellon

Main idea - SpikeM



- 1. **Un**-informed bloggers (uninformed about rumor)
- 2. External shock at time n_b (e.g, breaking news)
- 3. Infection (word-of-mouth)



Time n=0 Time $n=n_b$ Time $n=n_b+1$

Infectiveness of a blog-post at age n:

β – Strength of infection (quality of news)

$f(n)$ – Decay function $\quad f(n) = \beta * n^{-1.5}$

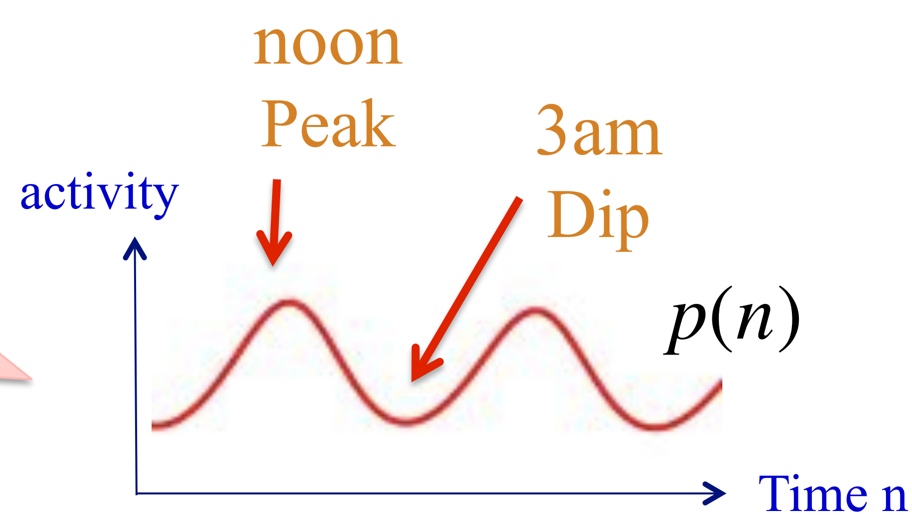
Details

SpikeM - with periodicity

- Full equation of SpikeM

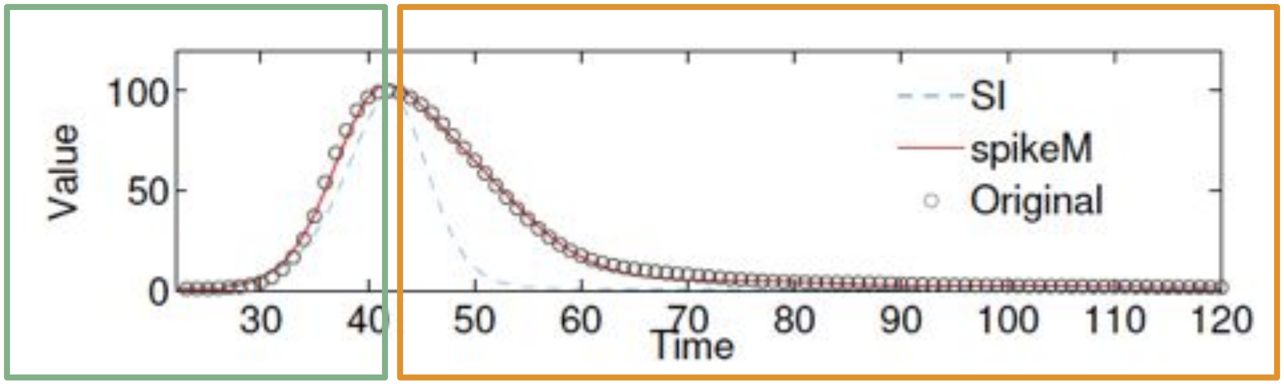
$$\Delta B(n+1) = \underbrace{p(n+1)}_{\text{Periodicity}} \cdot \left[U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot f(n+1-t) + \varepsilon \right]$$

Bloggers change their activity over time (e.g., daily, weekly, yearly)



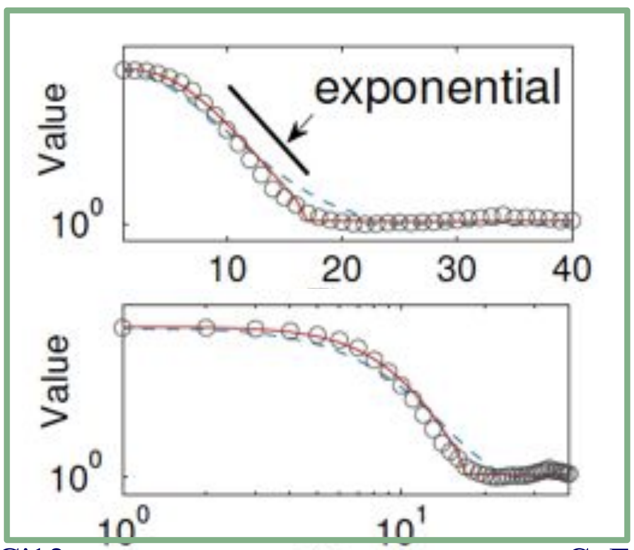
Details

- Analysis – exponential rise and power-raw fall



Lin-log

Log-log



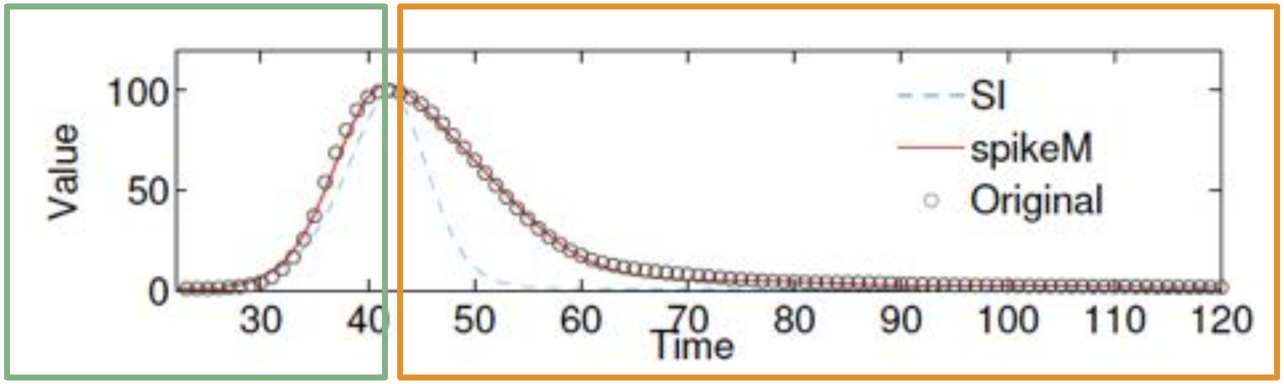
Rise-part

SI -> exponential
SpikeM -> exponential

Carnegie Mellon

Details

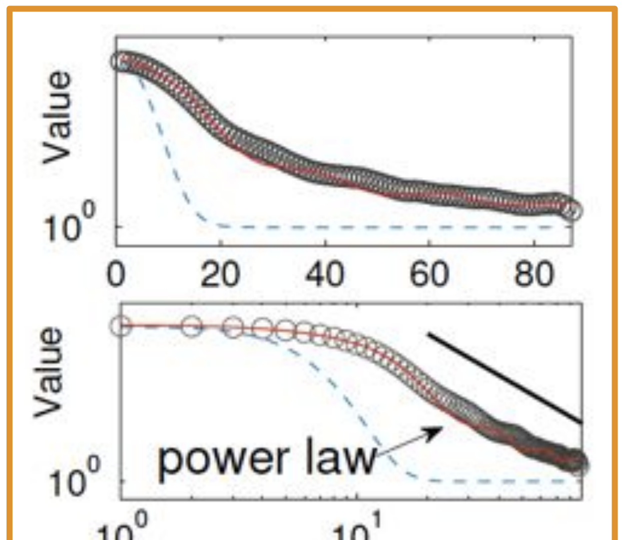
- Analysis – exponential rise and power-raw fall



Fall-part

❌ SI -> exponential

SpikeM -> power law

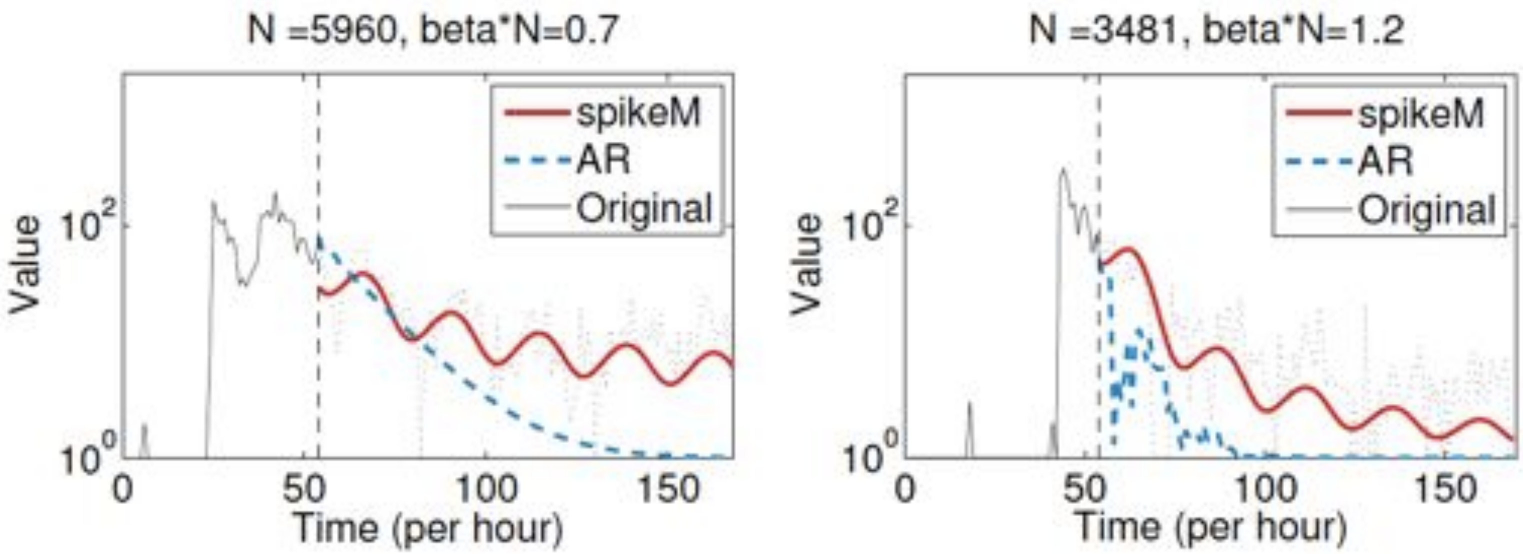


Lin-log

Log-log

Tail-part forecasts

- **SpikeM** can capture tail part



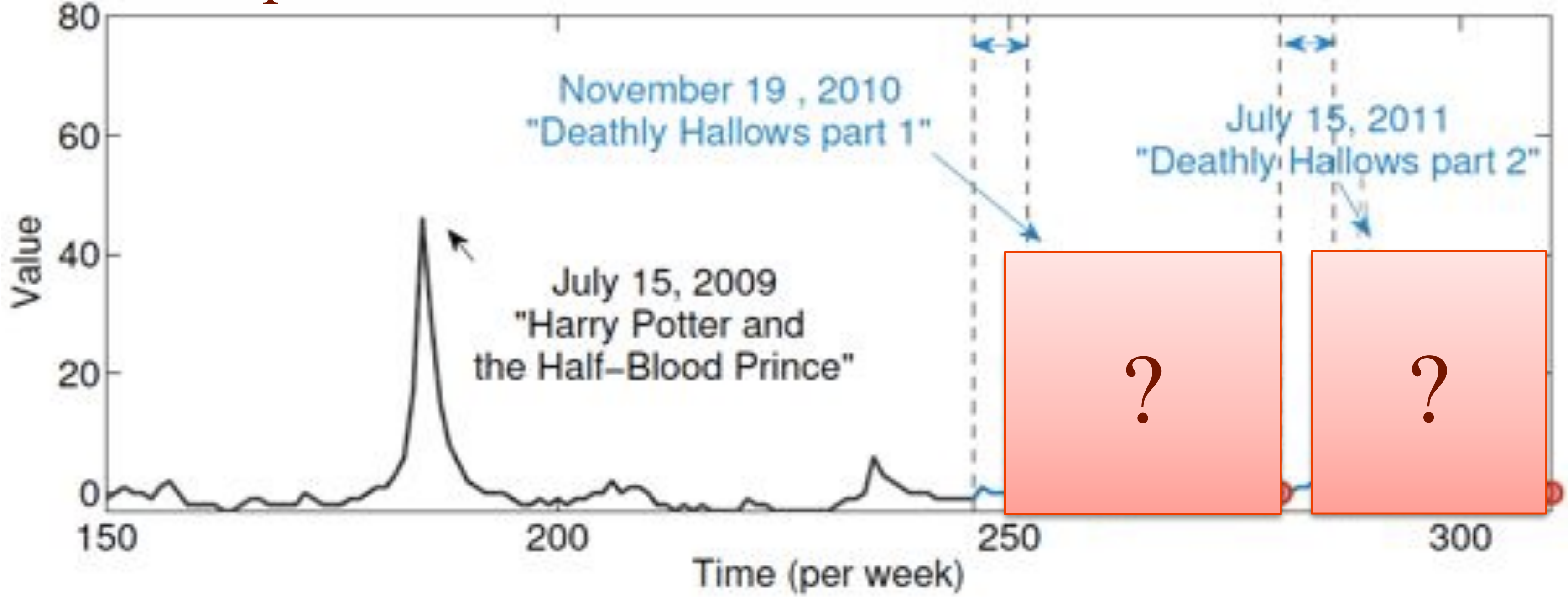
Carnegie Mellon

"What-if" forecasting

(1) First spike

(2) Release date

(3) Two weeks before release



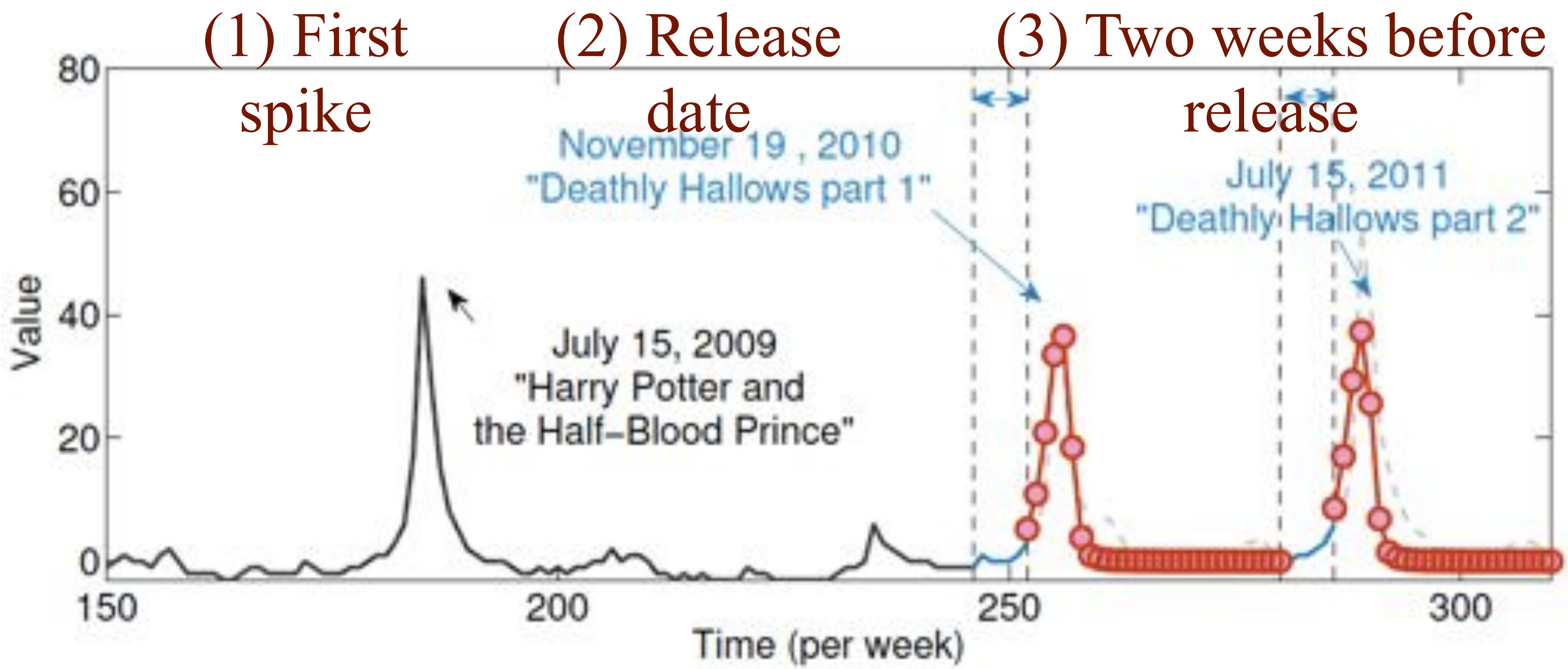
e.g., given (1) first spike,

(2) release date of two sequel movies

(3) access volume before the release date

"What-if" forecasting

–SpikeM can forecast not only tail-part, but also **rise-part**!



- **SpikeM** can forecast shape of upcoming spikes

Roadmap

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief propagation
  - Spike analysis
  - Tensors
- Problem#3: Scalability -PEGASUS
- Conclusions

Scalability

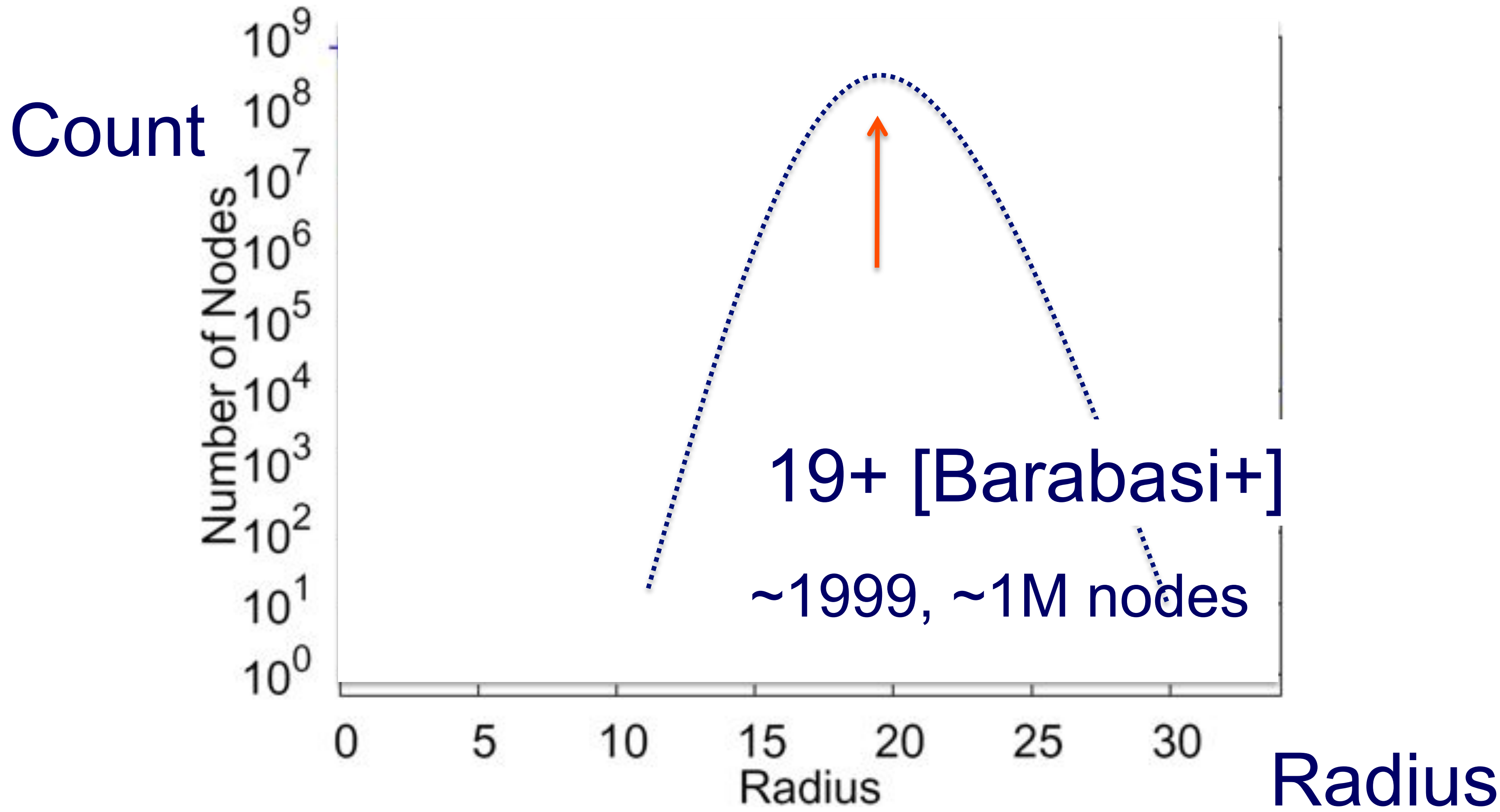
- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, *"Web Search for a Planet: The Google Cluster Architecture"* IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD'07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?
-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

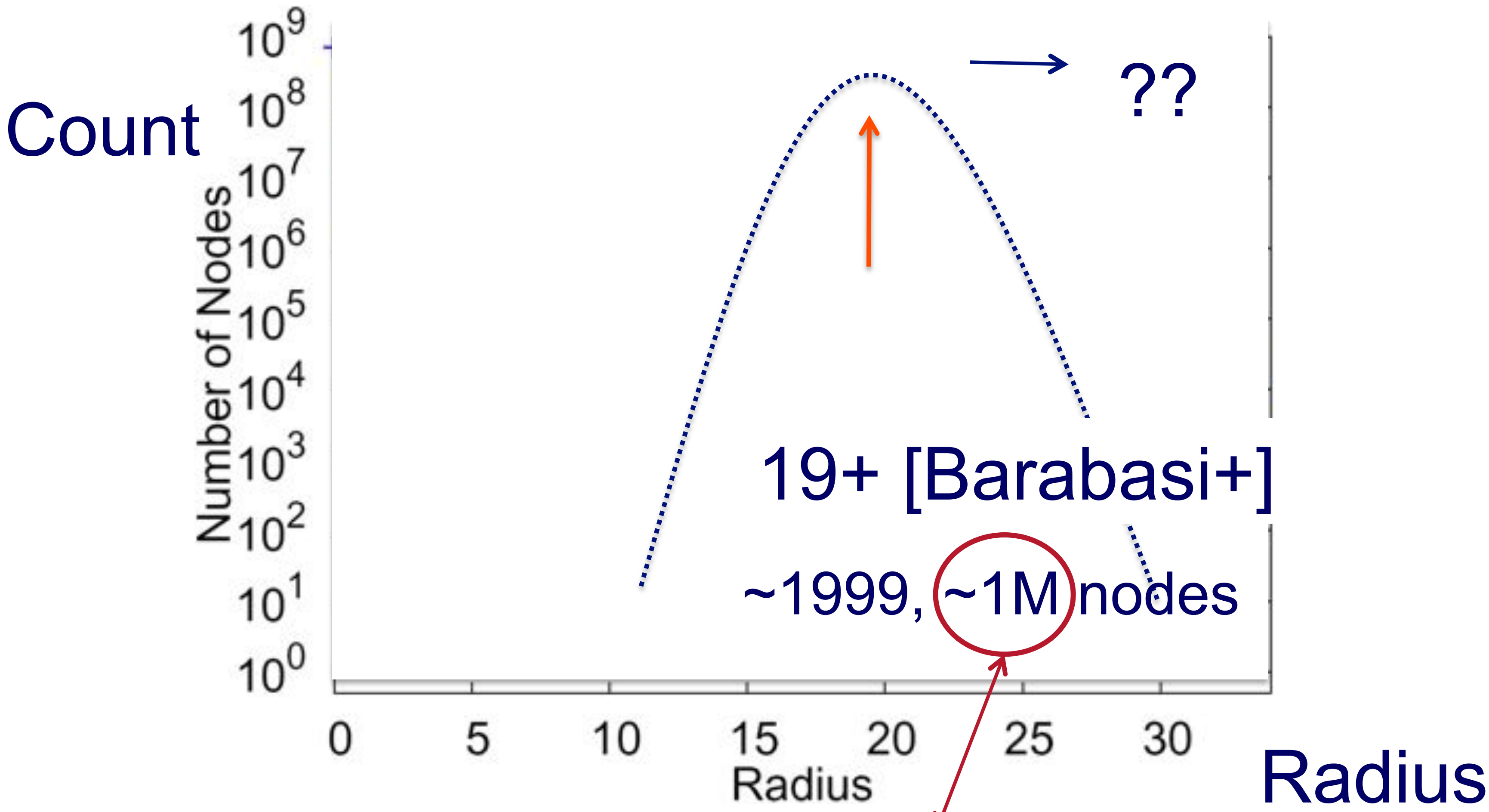
Roadmap – Algorithms & results

	Centralized	Hadoop/ PEGASUS
Degree Distr.	old	old
Pagerank	old	old
Diameter/ANF	old	**HERE**
Conn. Comp	old	**HERE**
Triangles	**done**	**HERE**
Visualization	**started**	

HADI for diameter estimation

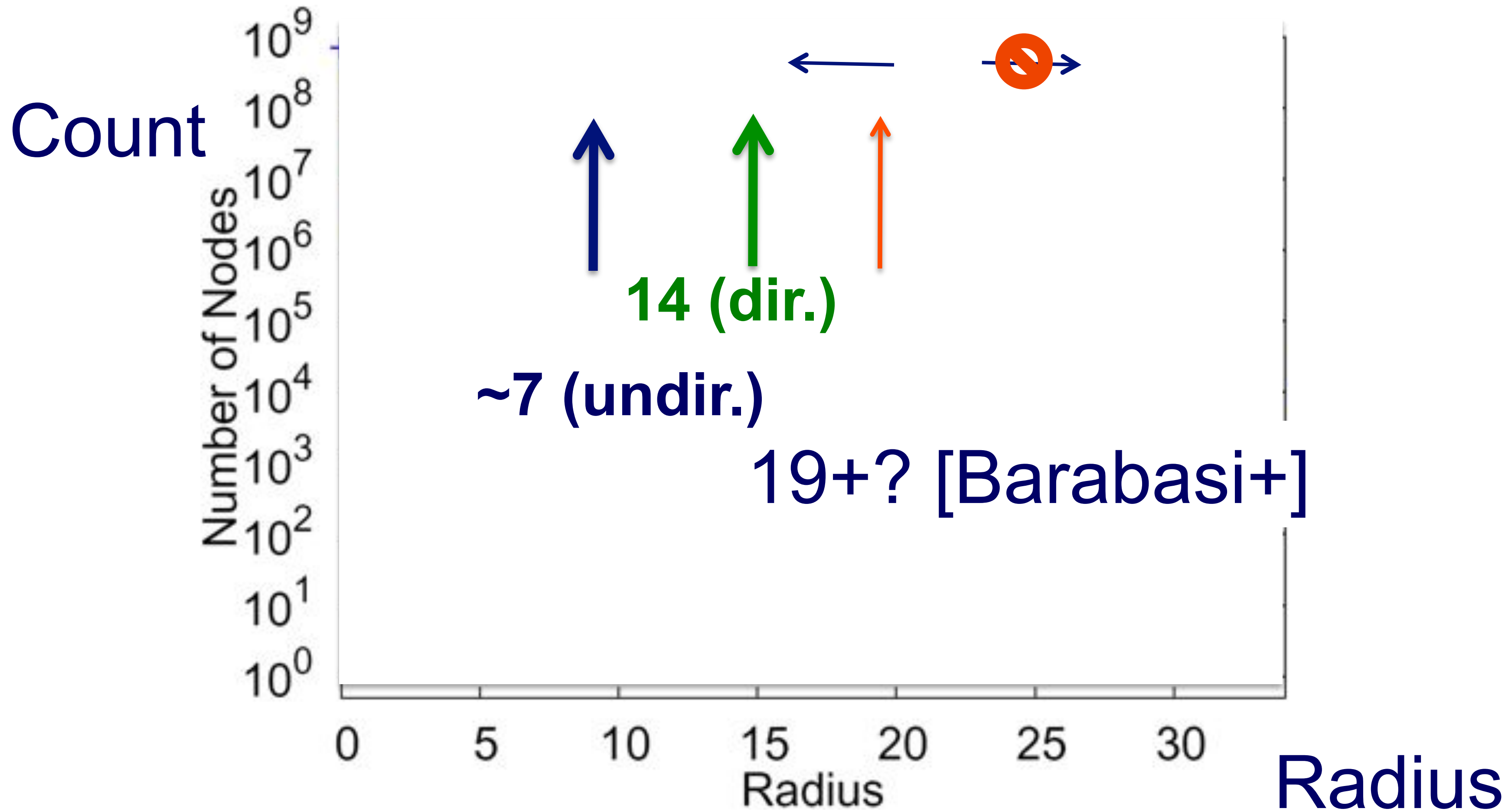
- *Radius Plots for Mining Tera-byte Scale Graphs* **U Kang**, Charalampos Tsourakakis, Ana Paula Appel, Christos Faloutsos, Jure Leskovec, SDM'10
- Naively: diameter needs **O(N**2)** space and up to O(N**3) time – **prohibitive** (N~1B)
- Our HADI: linear on E (~10B)
  - Near-linear scalability wrt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


Count
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
Number of Nodes
19+ [Barabasi+]
~1999, ~1M nodes
0
5
10
15
20
25
30
Radius
Radius



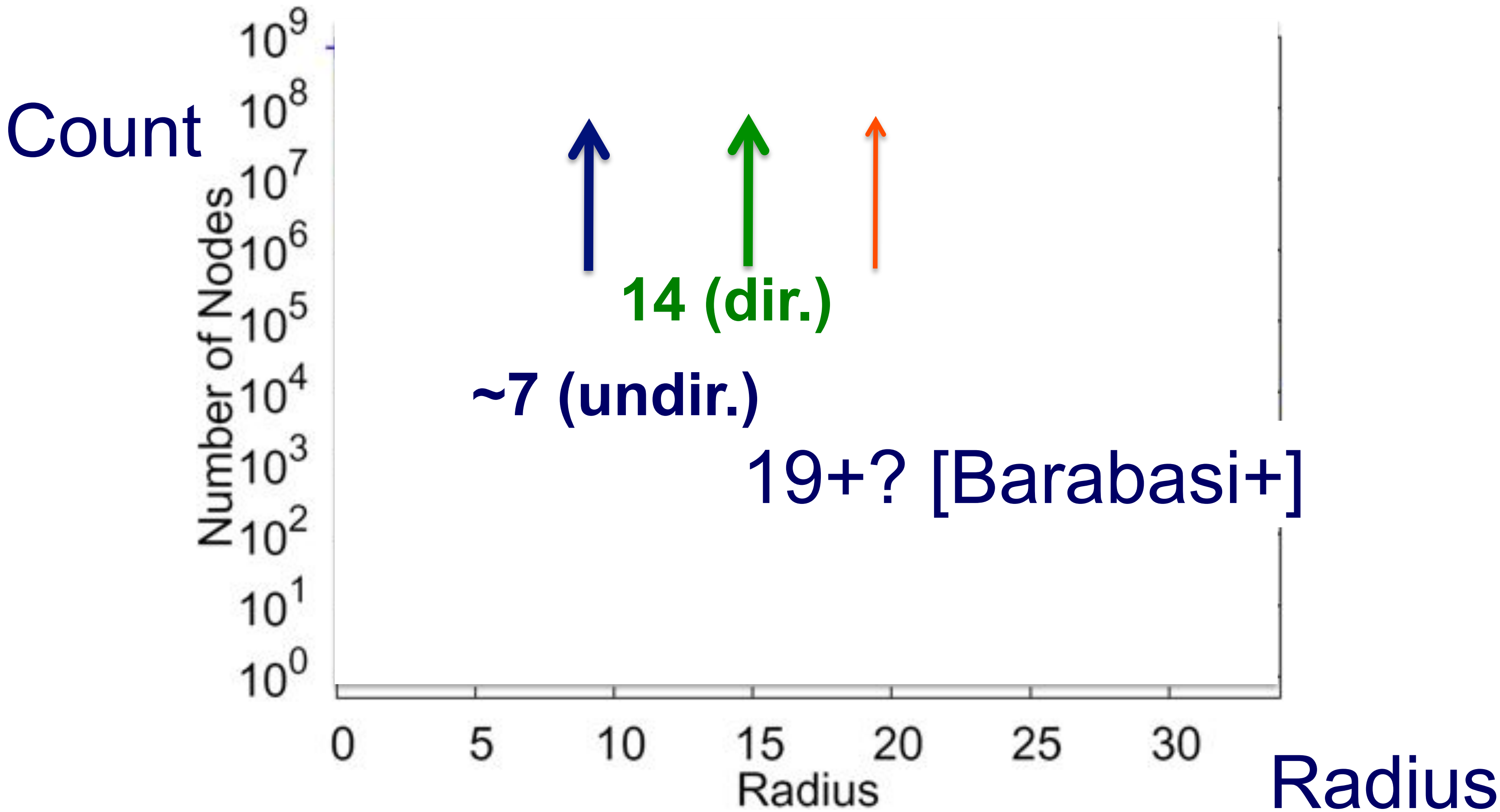
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

- Largest publicly available graph ever studied.



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

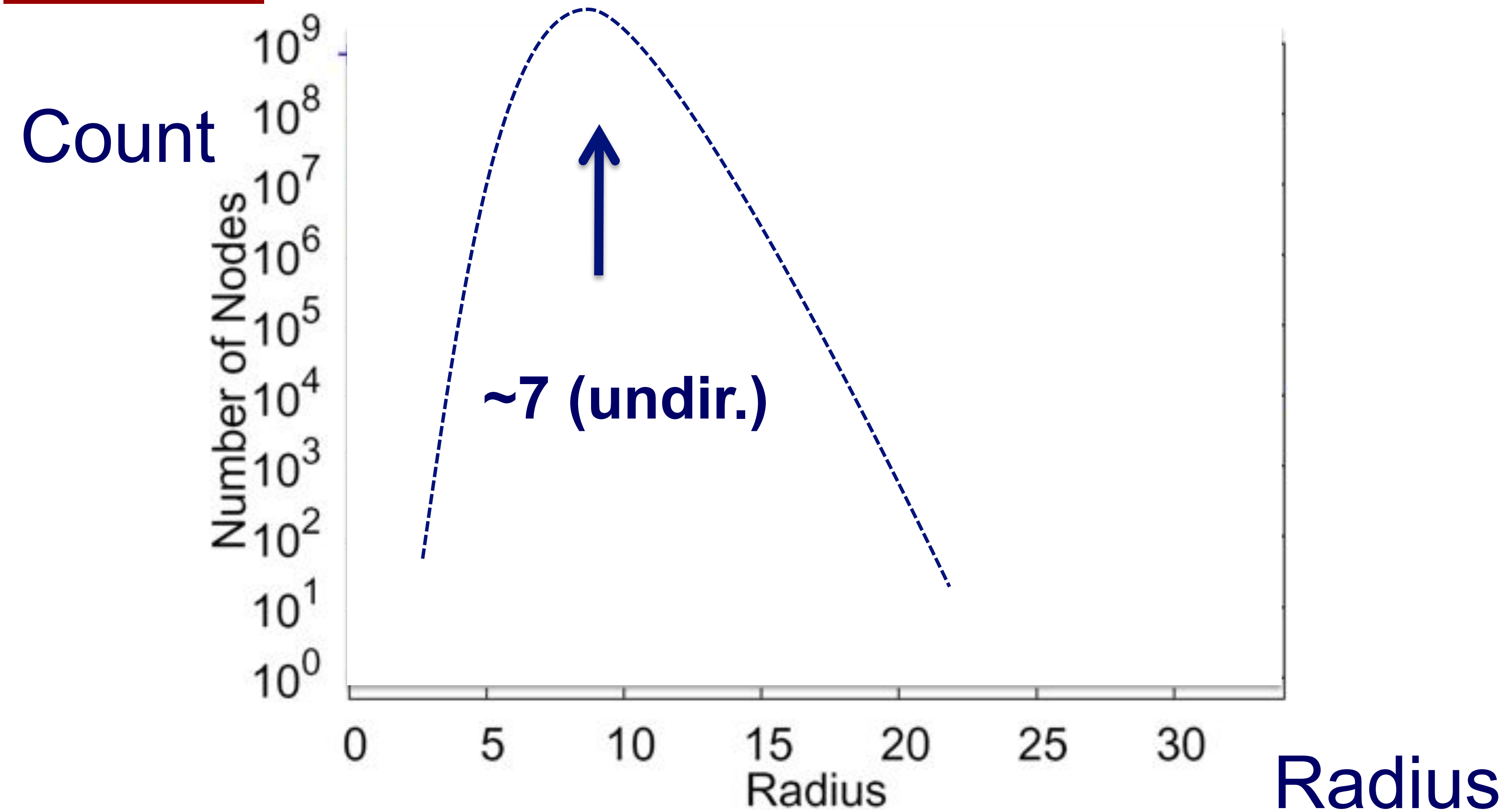
- Largest publicly available graph ever studied.



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

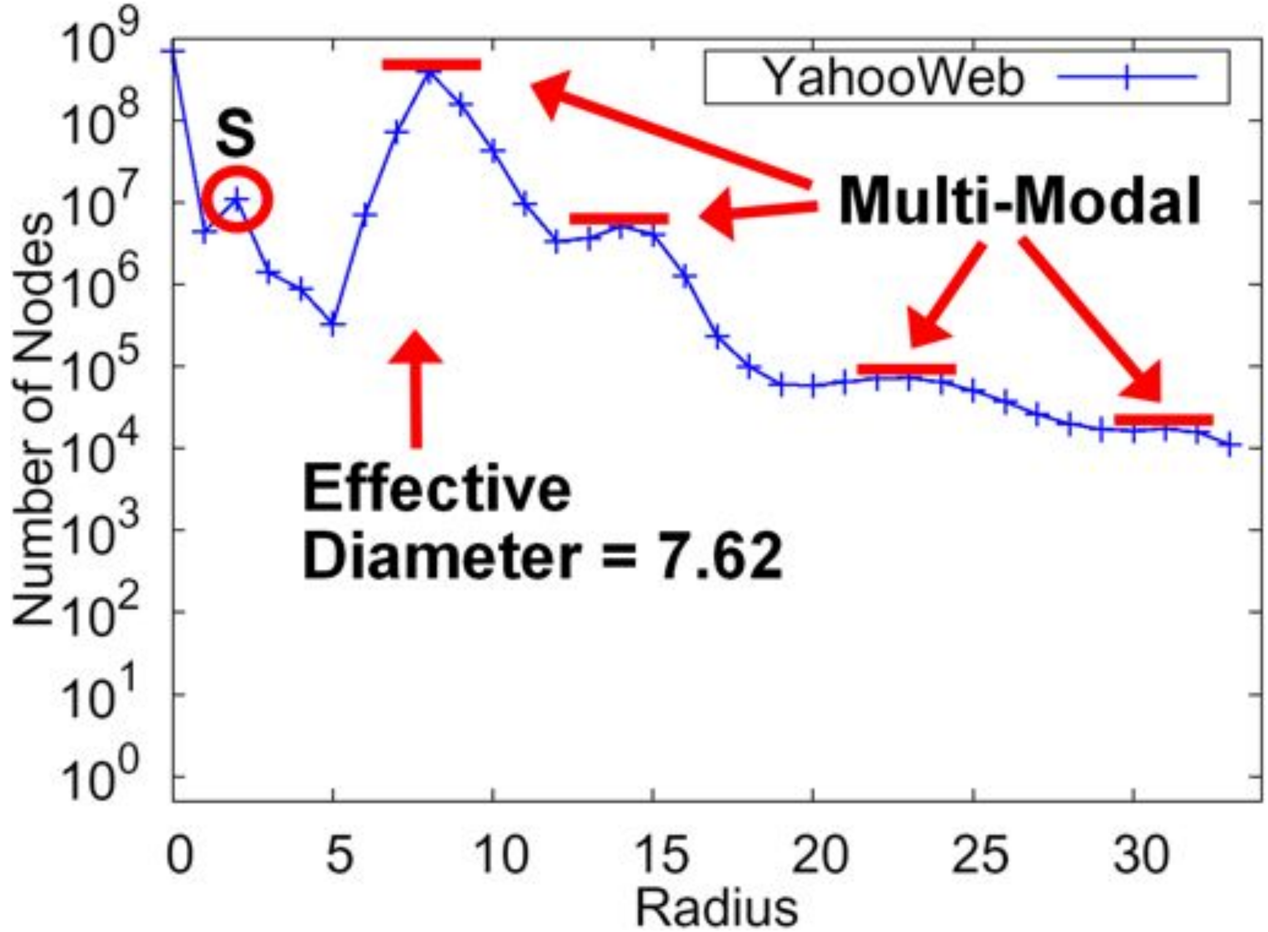
- 7 degrees of separation (!)
- Diameter: shrunk

Carnegie Mellon



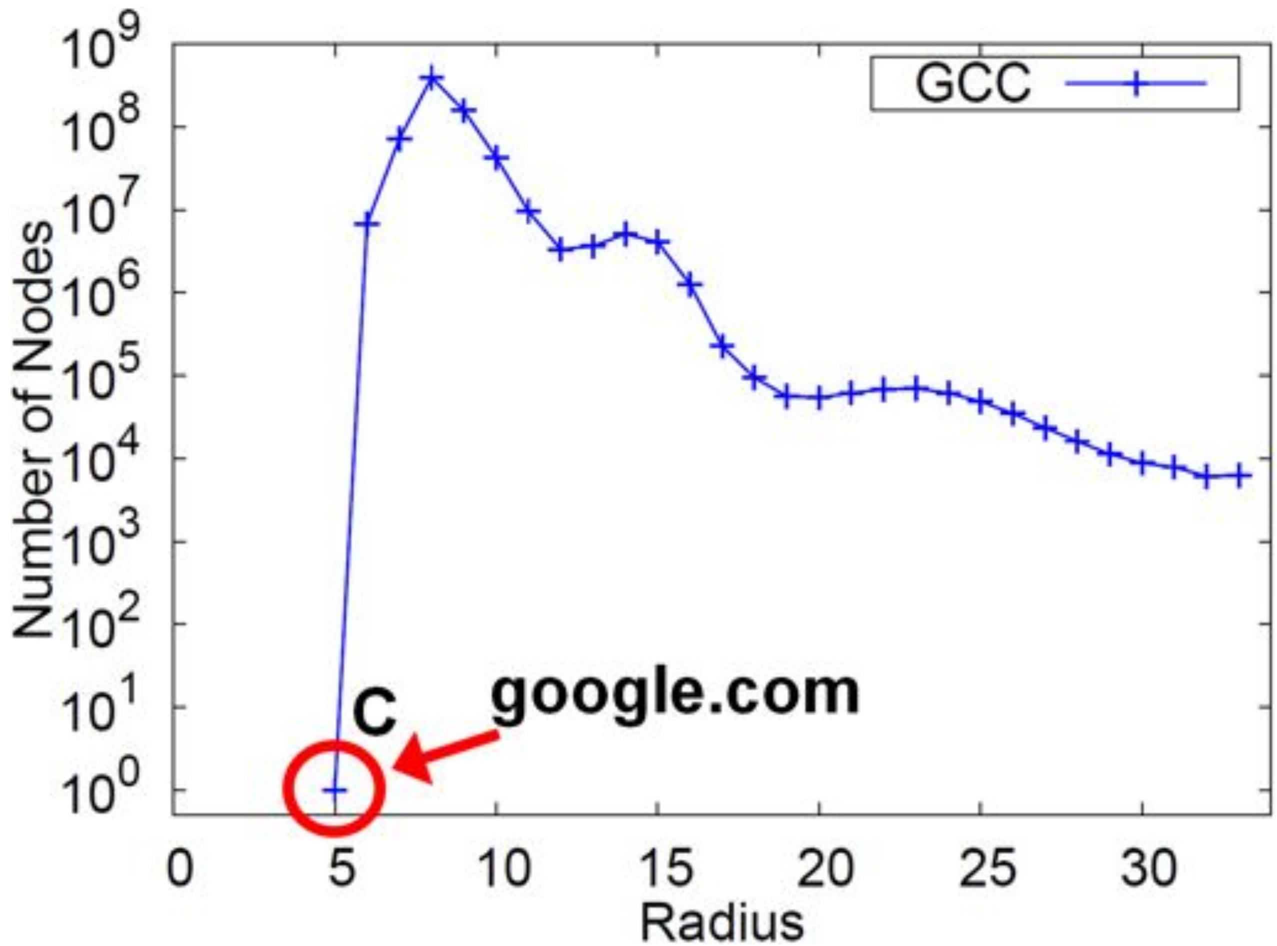
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

Q: Shape?

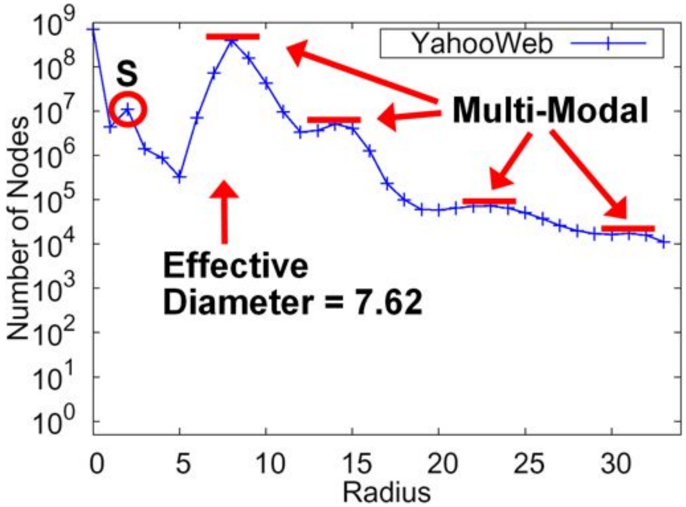


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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



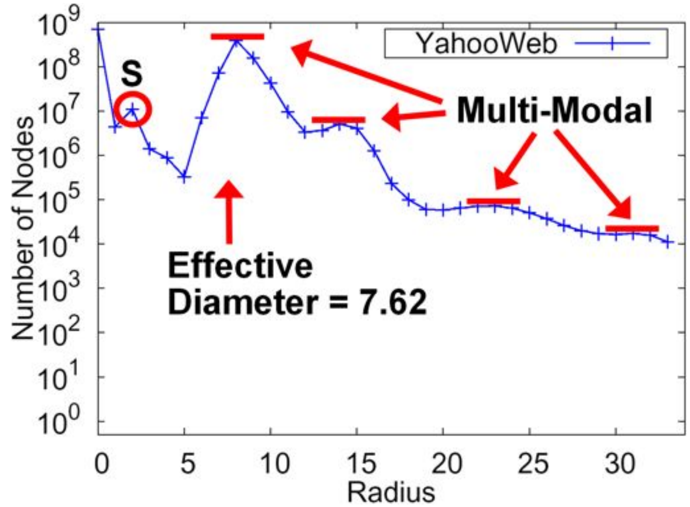
Radius Plot of **GCC** of YahooWeb.



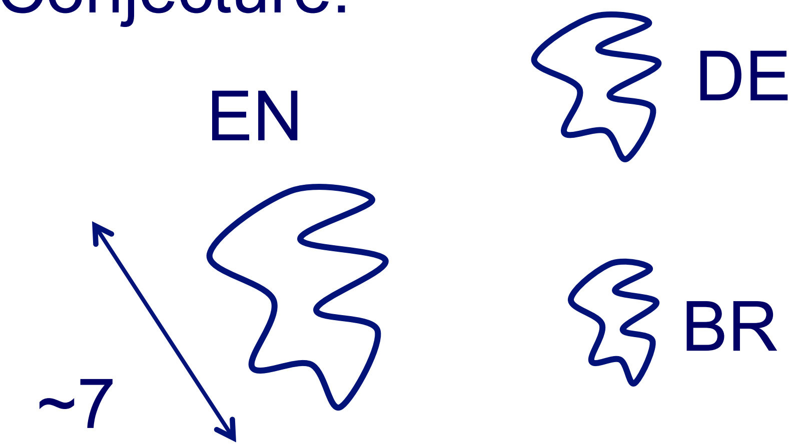
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

Carnegie Mellon



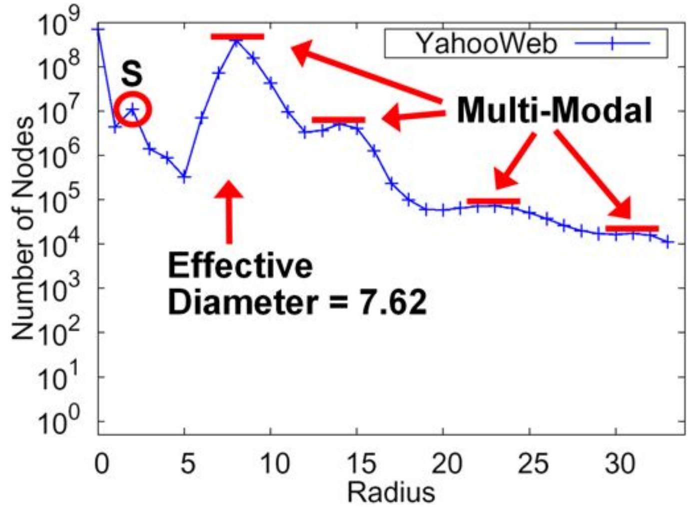
Conjecture:



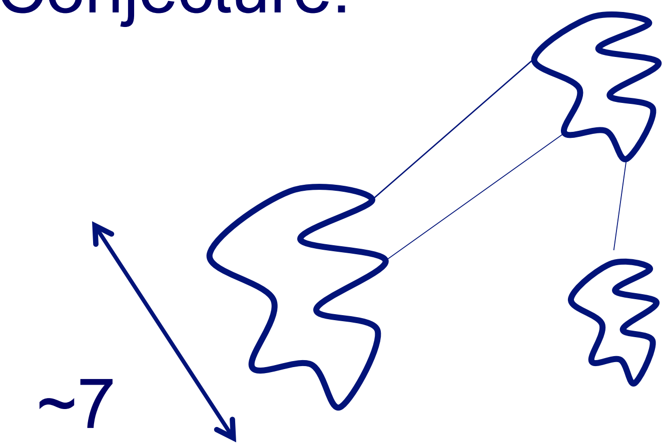
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

Carnegie Mellon



Conjecture:

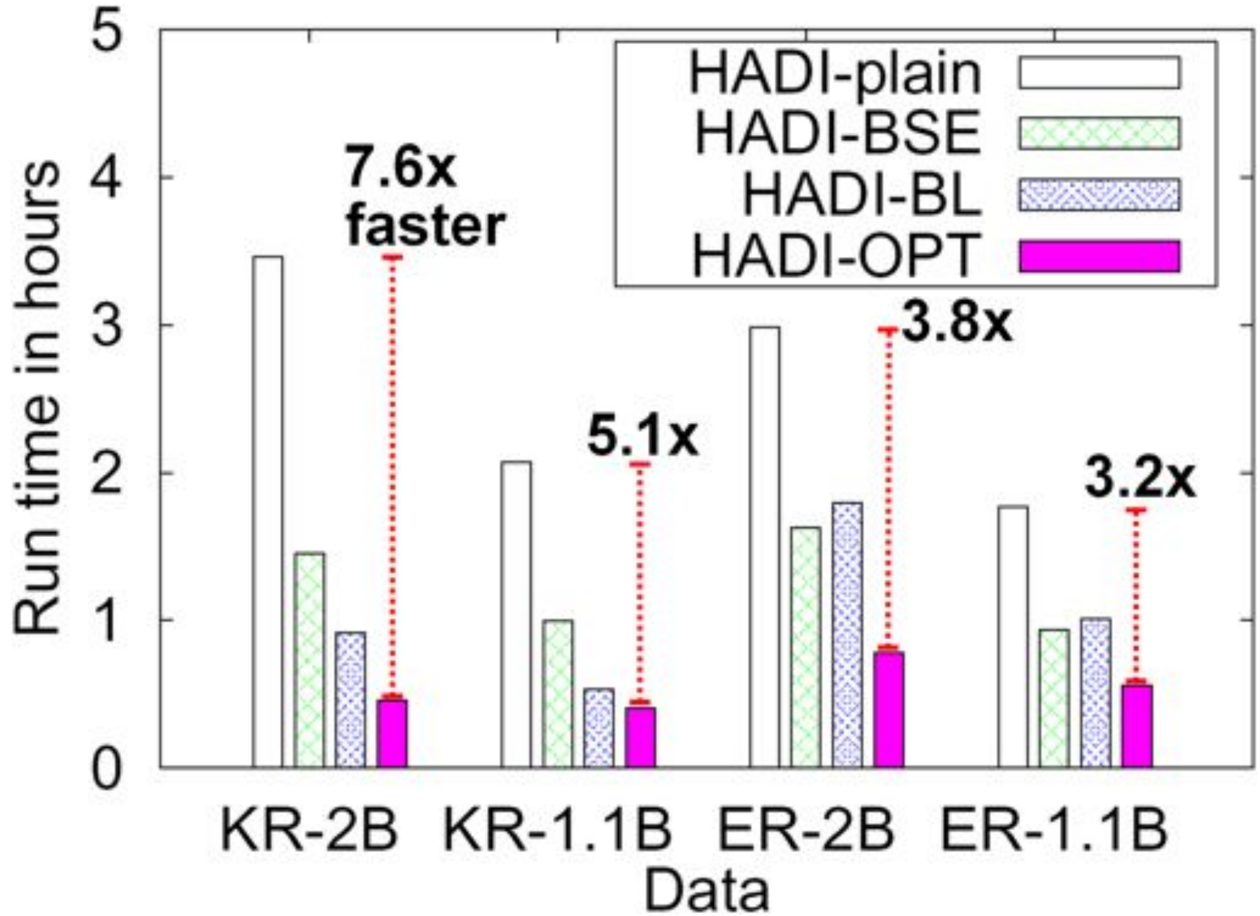


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

Carnegie Mellon

details



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

Carnegie Mellon

Roadmap – Algorithms & results

	Centralized	Hadoop/ PEGASUS
Degree Distr.	old	old
Pagerank	old	old
Diameter/ANF	old	**HERE**
Conn. Comp	old	**HERE**
Triangles		**HERE**
Visualization	**started**	

Carnegie Mellon

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).

details

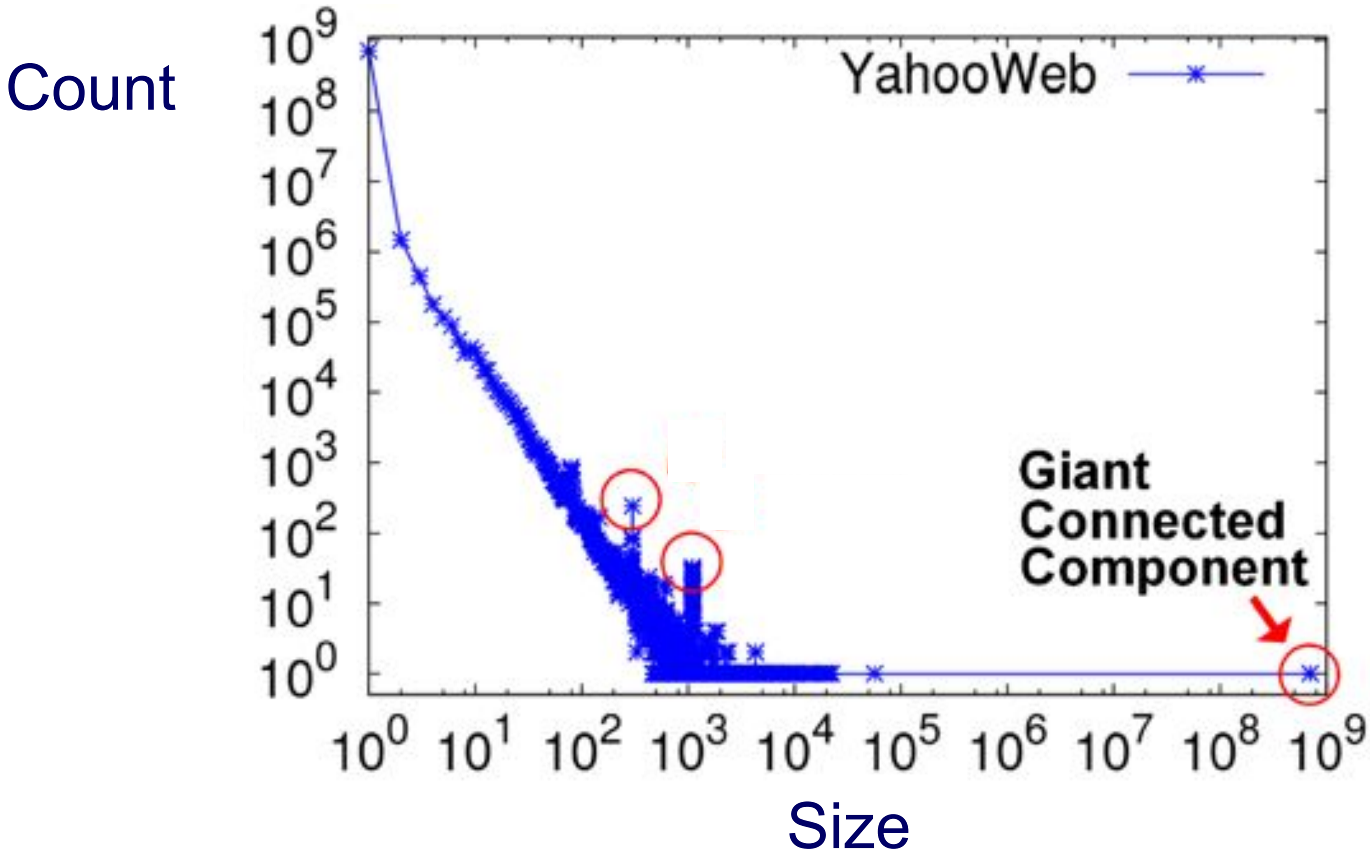
Generalized Iterated Matrix Vector Multiplication (GIMV)

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

Matrix – vector Multiplication (iterated)

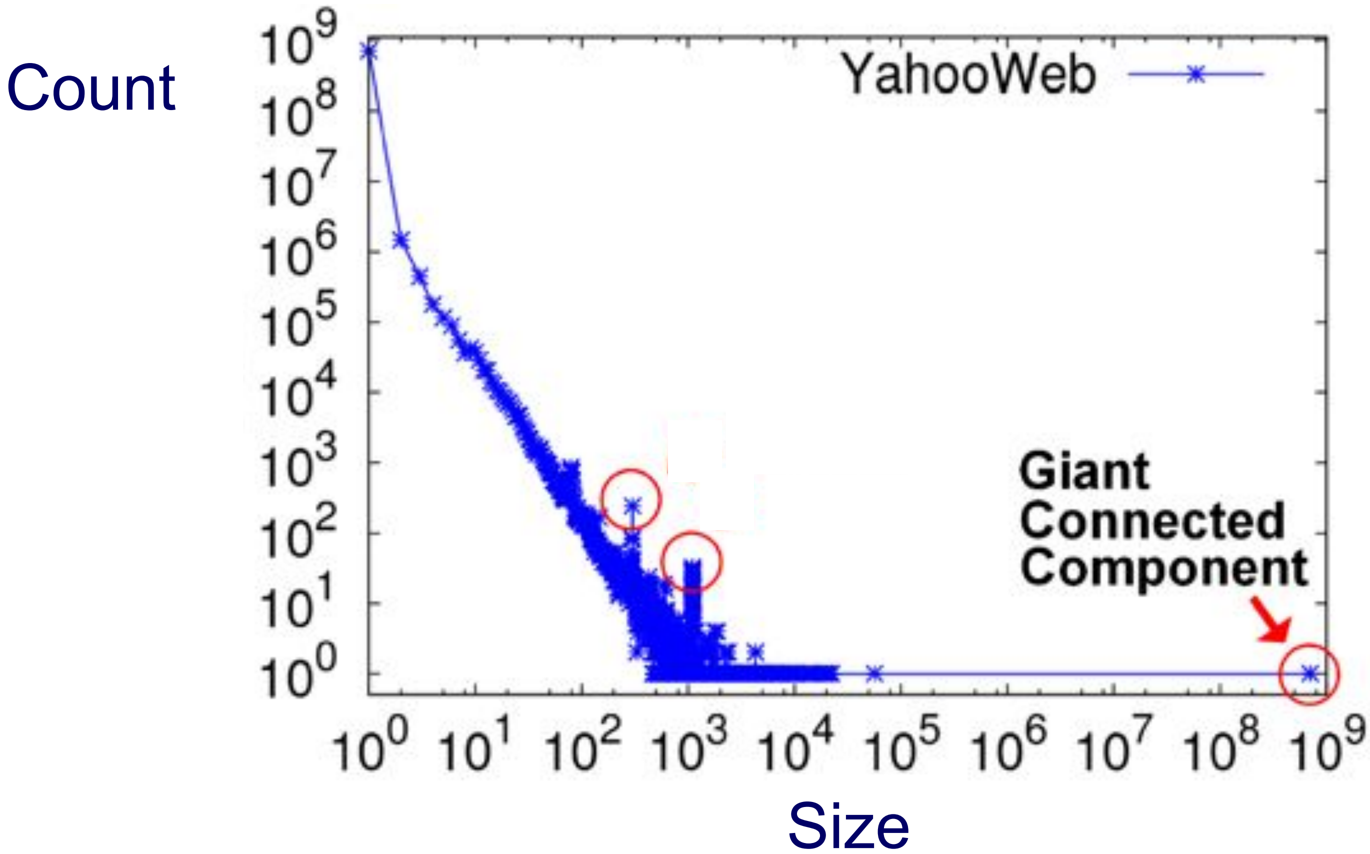
Example: GIM-V At Work

- Connected Components – 4 observations:



Example: GIM-V At Work

- Connected Components



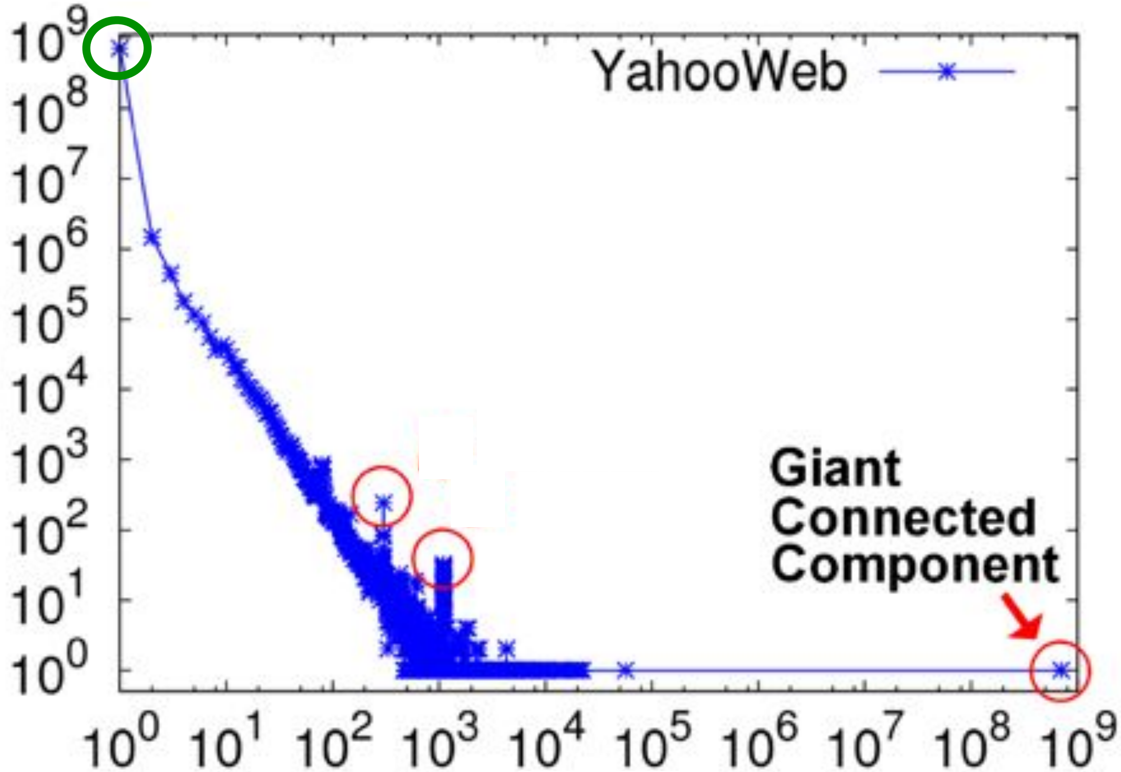
1) 10K x larger than next

Example: GIM-V At Work

- Connected Components

Count

2) ~0.7B singleton nodes



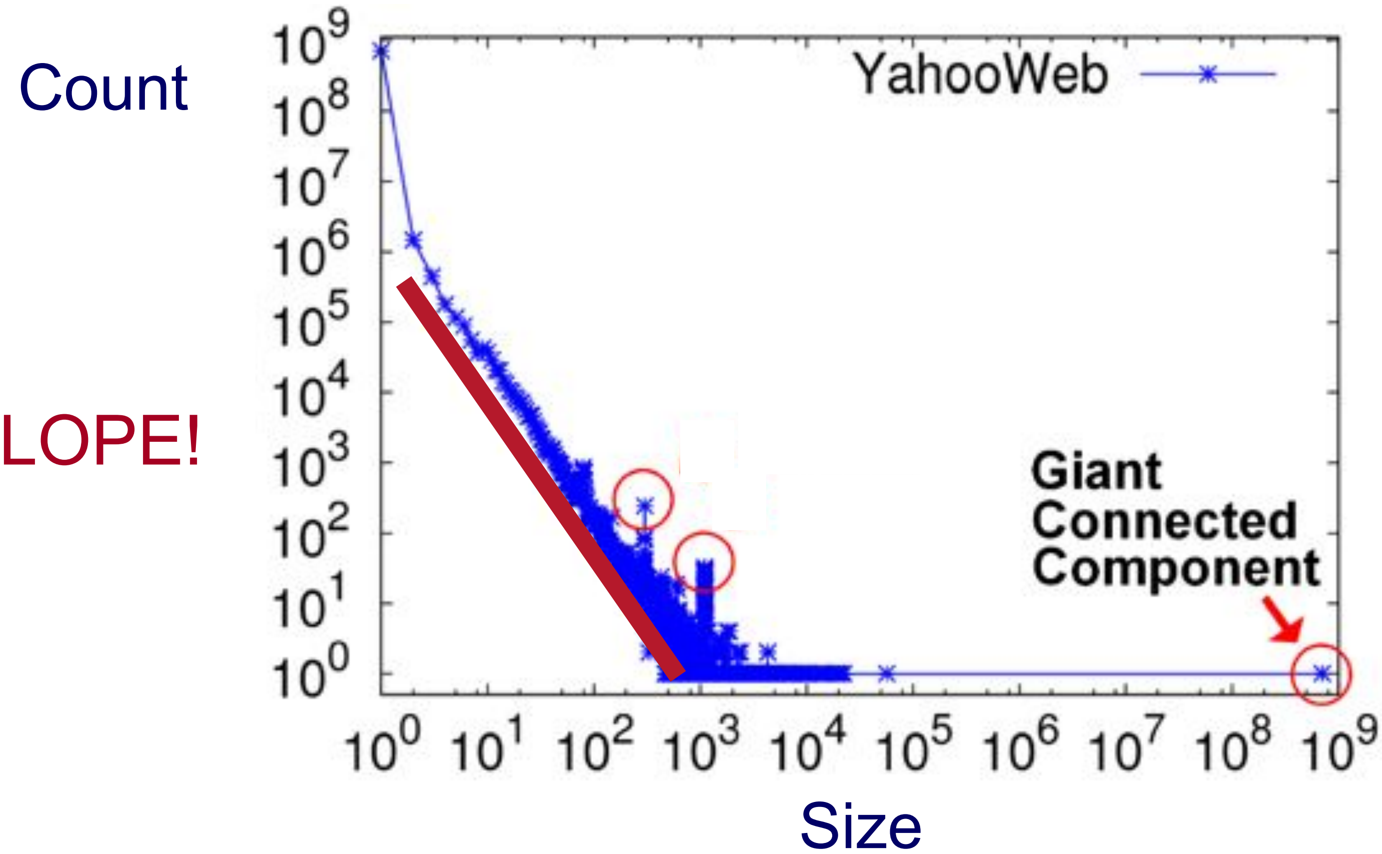
Size

Carnegie Mellon

Example: GIM-V At Work

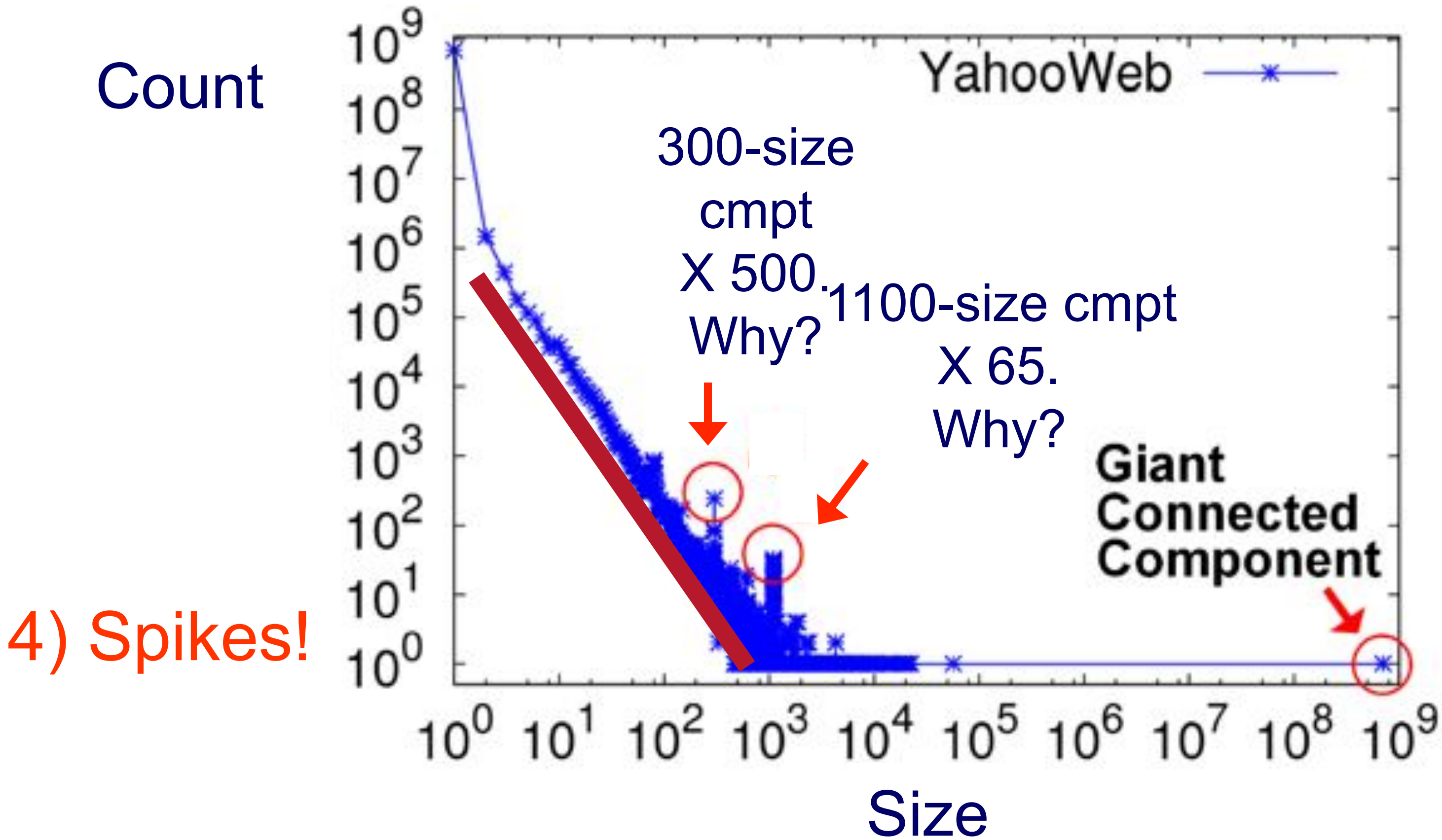
- Connected Components

3) SLOPE!



Example: GIM-V At Work

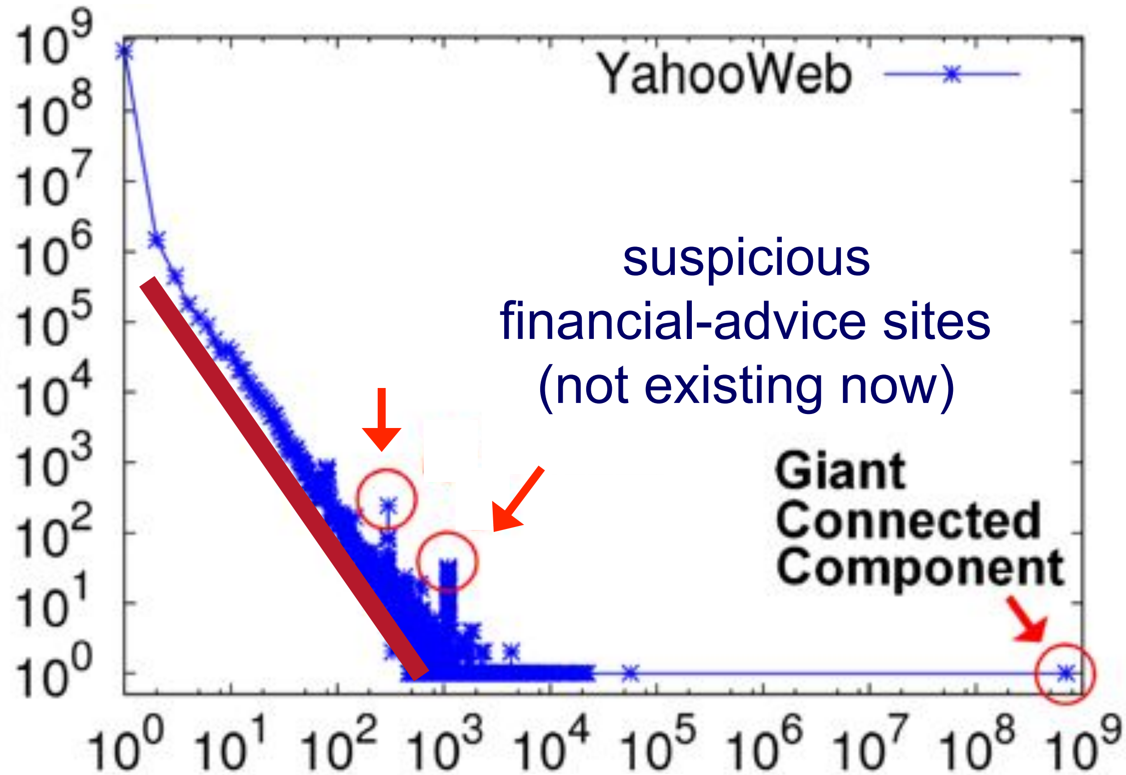
- Connected Components



Example: GIM-V At Work

- Connected Components

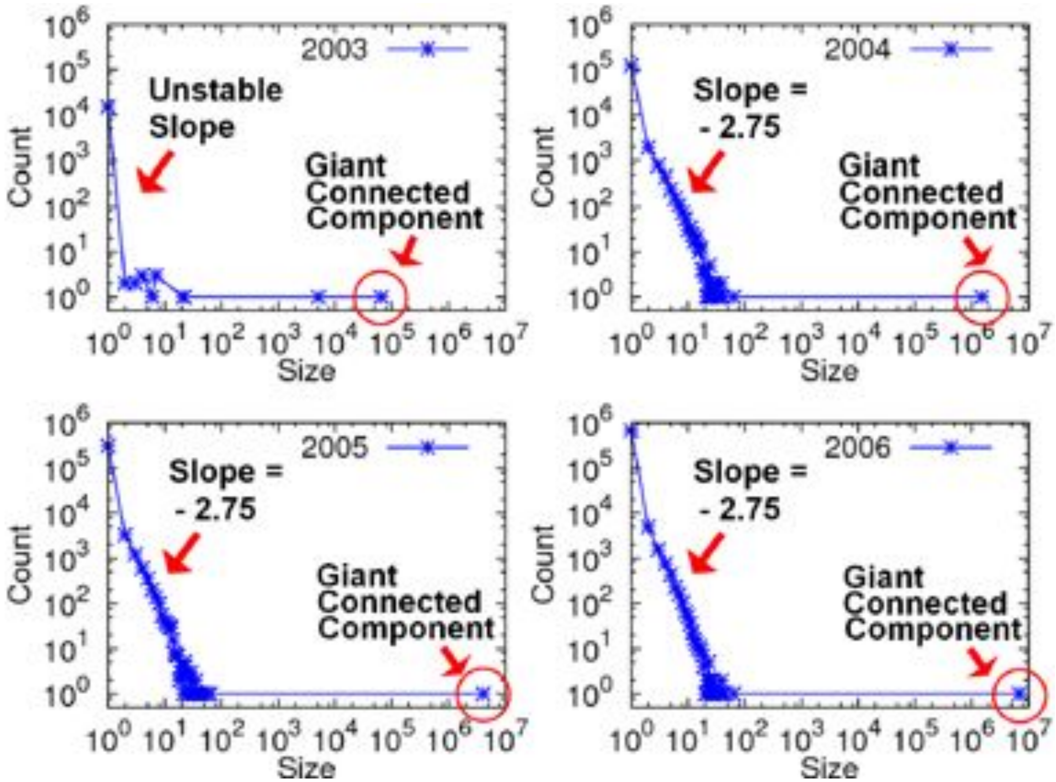
Count



Size

GIM-V At Work

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



Stable tail slope after the gelling point

Roadmap

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

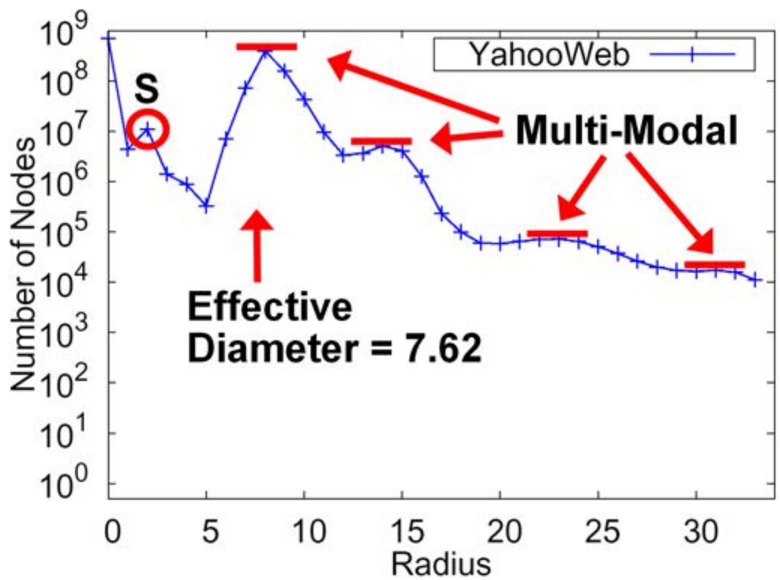
OVERALL CONCLUSIONS – low level:

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

Carnegie Mellon

OVERALL CONCLUSIONS – high level

- **BIG DATA: Large** datasets reveal 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)

Carnegie Mellon

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

Carnegie Mellon

Project info

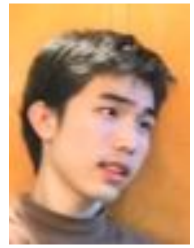
www.cs.cmu.edu/~pegasus

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

Cast

Akoglu, Leman

Beutel, Alex



Chau, Polo

Kang, U

Koutra, Danae

McGlohon, Mary

Prakash, Aditya

Papalexakis, Vagelis

Tong, Hanghang

OVERALL CONCLUSIONS – high level

- **BIG DATA: Large** datasets reveal patterns/ outliers that are invisible otherwise