Carnegie Mellon

15-826: Multimedia Databases and Data Mining

Lecture #28: Graph mining - patterns

Christos Faloutsos

Carnegie Mellon

Must-read Material

- [Graph minining textbook] Deepayan Chakrabarti and Christos Faloutsos *Graph Mining: Laws, Tools and Case Studies*, Morgan Claypool, 2012
  – Part I (patterns)

Carnegie Mellon

Must-read Material

- Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos, On Power-Law Relationships of the Internet Topology, SIGCOMM 1999.
- R. Albert, H. Jeong, and A.-L. Barabasi, Diameter of the World Wide Web Nature, 401, 130-131 (1999).
- Reka Albert and Albert-Laszlo Barabasi Statistical mechanics of complex networks, Reviews of Modern Physics, 74, 47 (2002).
- Jure Leskovec, Jon Kleinberg, Christos Faloutsos Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations, KDD 2005, Chicago, IL, USA

Carnegie Mellon

Must-read Material (cont' d)

- D. Chakrabarti and C. Faloutsos, Graph Mining: Laws, Generators and Algorithms, in ACM Computing Surveys, 38(1), 2006

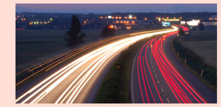
CarnegieMellon

Main outline

- Introduction
- Indexing
- Mining
  - Graphs – patterns
  - Graphs – generators and tools
  - Association rules
  - …

CarnegieMellon

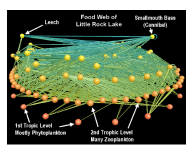
Outline



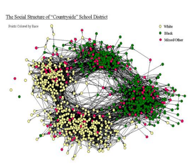
- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Scalability
- Conclusions

CarnegieMellon

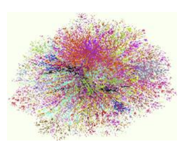
Graphs - why should we care?



Food Web
[Martinez '91]



Friendship Network
[Moody '01]

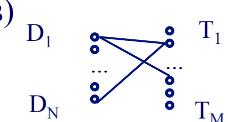


Internet Map
[lumeta.com]

CarnegieMellon

Graphs - why should we care?

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



- web: hyper-text graph
- ... and more:

Carnegie Mellon

Graphs - why should we care?

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

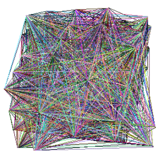
Carnegie Mellon

Outline

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

Carnegie Mellon

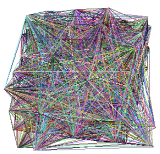
Problem #1 - network and graph mining



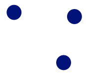
- What does the Internet look like?
- What does FaceBook look like?
- What is 'normal'/'abnormal'?
- which patterns/laws hold?

Carnegie Mellon

Problem #1 - network and graph mining

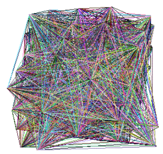


- What does the Internet look like?
- What does FaceBook look like?
- What is 'normal'/'abnormal'?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**



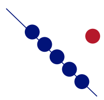
Carnegie Mellon

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…

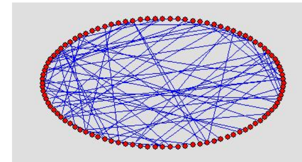


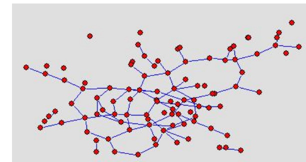
Carnegie Mellon

Are real graphs random?

- random (Erdos-Renyi) graph – 100 nodes, avg degree = 2
- before layout
- after layout
- No obvious patterns

(generated with: pajek http://vlado.fmf.uni-lj.si/pub/networks/pajek/)





Carnegie Mellon

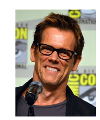
Graph mining

- Are real graphs random?

Carnegie Mellon

Laws and patterns

- Are real graphs random?
- A: NO!!
  - Diameter ('6 degrees', 'Kevin Bacon')
  - in- and out- degree distributions
  - other (surprising) patterns



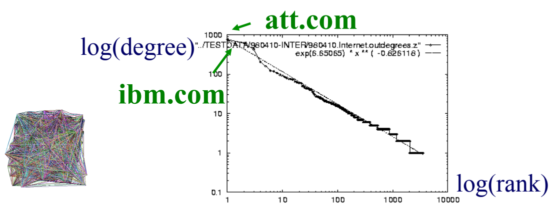
- So, let's look at the data

Carnegie Mellon

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

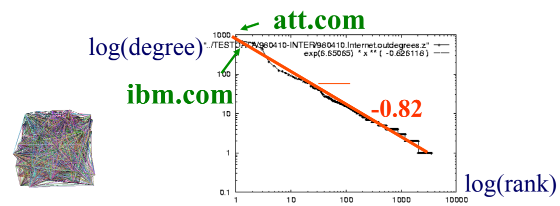


Carnegie Mellon

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains

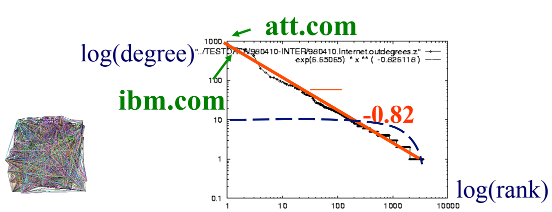


Carnegie Mellon

Solution# S.1

- Q: So what?

internet domains

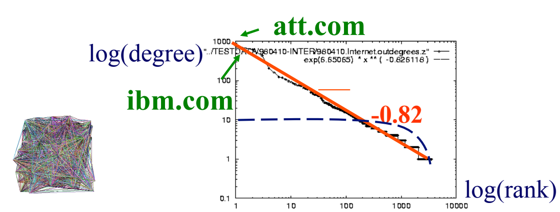


Carnegie Mellon

Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs: = friends of friends (F.O.F.)

internet domains

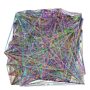
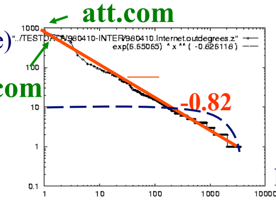


Carnegie Mellon

Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs: 100^2 * N= 10 Trillion

= friends of friends (F.O.F.)

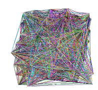
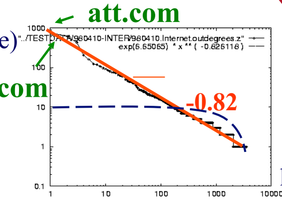



Carnegie Mellon

Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs: 100^2 * N= 10 Trillion

= friends of friends (F.O.F.)

Carnegie Mellon

Gaussian trap

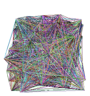
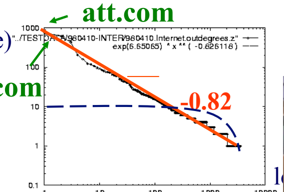
Solution# S.1

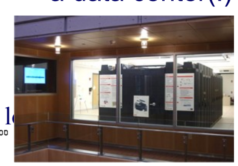
- Q: So what?
- A1: # of two-step-away pairs: O(d_max ^2) ~ 10M^2

= friends of friends (F.O.F.)

⇩

~0.8PB ->
a data center(!)



DCO @ CMU

Carnegie Mellon

Gaussian trap

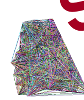
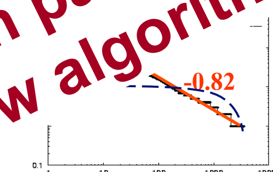
Solution# S.1

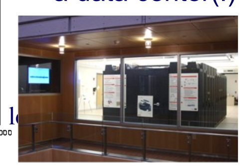
- Q: So what?
- A1: # of two-step-aw ?) ~ 10M^2

⇩

~0.8PB ->
a data center(!)

Such patterns ->
New algorithms

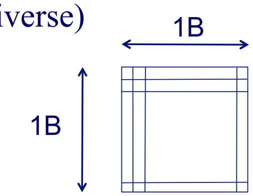





Carnegie Mellon

Observation – big-data:

- O(N^2) algorithms are ~intractable - N=1B
- N^2 seconds = 31B years (>2x age of universe)

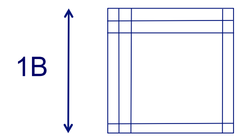


15-826 (c) C. Faloutsos, 2016 25

Carnegie Mellon

Observation – big-data:

- O(N^2) algorithms are ~intractable - N=1B
- N^2 seconds = ~~31B~~ 31M years
- 1,000 machines

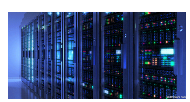


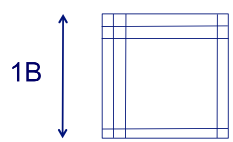
15-826 (c) C. Faloutsos, 2016 26

Carnegie Mellon

Observation – big-data:

- O(N^2) algorithms are ~intractable - N=1B
- N^2 seconds = ~~31B~~ 31K years
- 1M machines





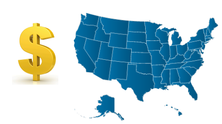
Google Y!

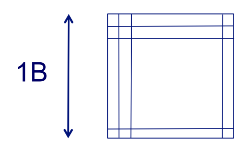
15-826 (c) C. Faloutsos, 2016 27

Carnegie Mellon

Observation – big-data:

- O(N^2) algorithms are ~intractable - N=1B
- N^2 seconds = ~~31B~~ 3 years
- 10B machines ~ $10Trillion





15-826 (c) C. Faloutsos, 2016 28

Carnegie Mellon

Observation – big-data:

- O(N^2) algorithms are ~intractable - N=1B

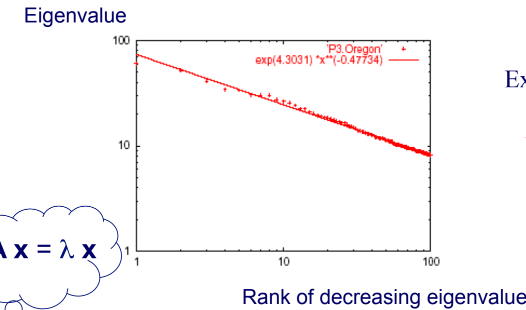
And parallelism might not help

- N^2 seconds = ~~31~~ 3 1B years
- 10B machines ~ $10Trillion

15-826 (c) C. Faloutsos, 2016 29

Carnegie Mellon

Solution# S.2: Eigen Exponent *E*



A x = λ x

Exponent = slope

E = -0.48

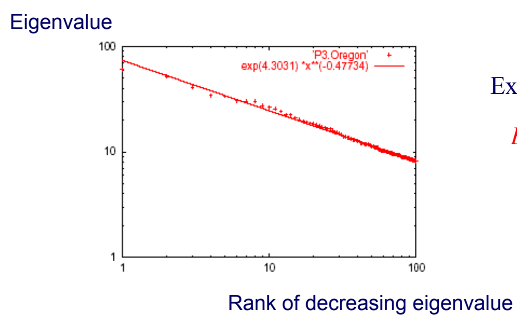
May 2001

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

15-826 (c) C. Faloutsos, 2016 30

Carnegie Mellon

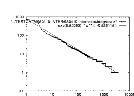
Solution# S.2: Eigen Exponent *E*



Exponent = slope

E = -0.48

May 2001



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

15-826 (c) C. Faloutsos, 2016 31

Carnegie Mellon

But:

How about graphs from other domains?

15-826 (c) C. Faloutsos, 2016 32

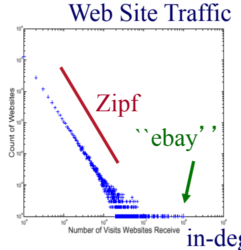
Carnegie Mellon

More power laws:

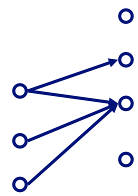
- web hit counts [w/ A. Montgomery]

Web Site Traffic

Count (log scale)



in-degree (log scale)



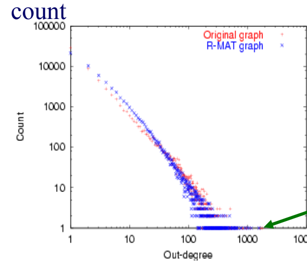
15-826 (c) C. Faloutsos, 2016 33

Carnegie Mellon

epinions.com

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

count



trusts-2000-people user

(out) degree

15-826 (c) C. Faloutsos, 2016 34

Carnegie Mellon

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'

15-826 (c) C. Faloutsos, 2016 35

Carnegie Mellon

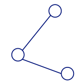
Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - ➡ Triangles
  - Weighted graphs
  - Time evolving graphs

15-826 (c) C. Faloutsos, 2016 36

Carnegie Mellon

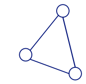
Solution# S.3: Triangle 'Laws'



- Real social networks have a lot of triangles

Carnegie Mellon

Solution# S.3: Triangle 'Laws'

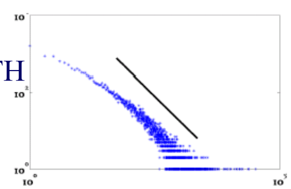


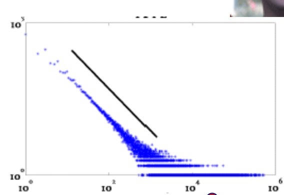
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

Carnegie Mellon

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

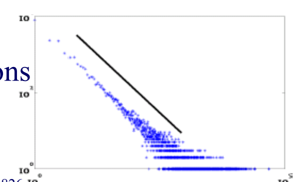
HEP-TH

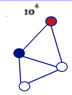




ASN

Epinions



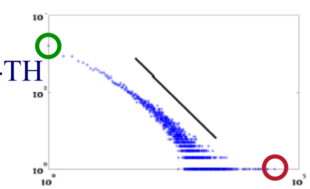


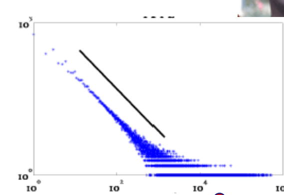
X-axis: # of participating triangles
Y: count (~ pdf)

Carnegie Mellon

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

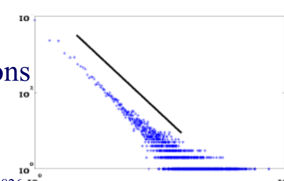
HEP-TH

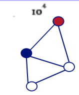




ASN

Epinions





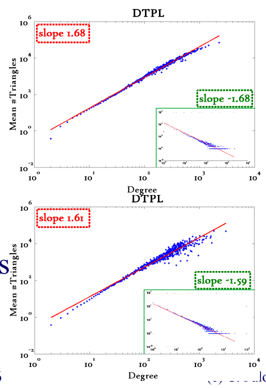
X-axis: # of participating triangles
Y: count (~ pdf)

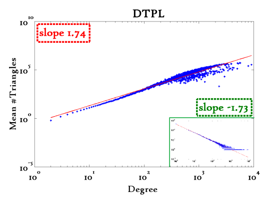
Carnegie Mellon

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

Reuters

SN





Epinions

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

Carnegie Mellon

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?

Carnegie Mellon

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!

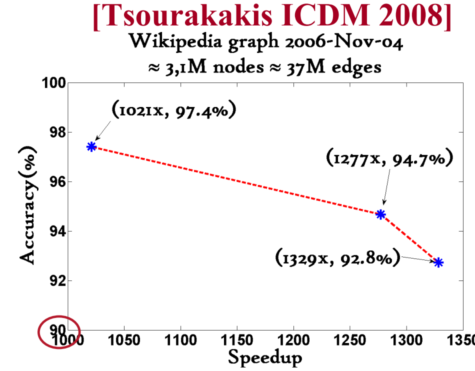
#triangles = 1/6 Sum (λ_i^3)

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

Carnegie Mellon

details

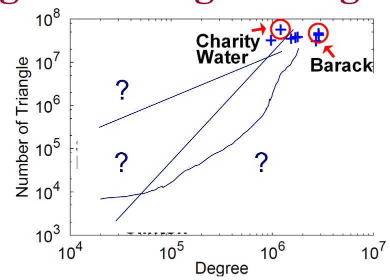
Triangle Law: Computations
[Tsourakakis ICDM 2008]



1000x+ speed-up, >90% accuracy

Carnegie Mellon

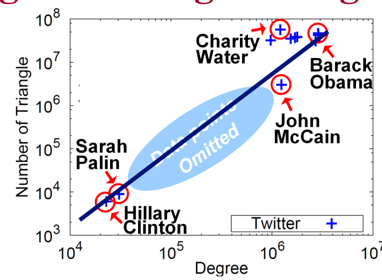
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)
[U Kang, Brendan Meeder, +, PAKDD'11]


Carnegie Mellon

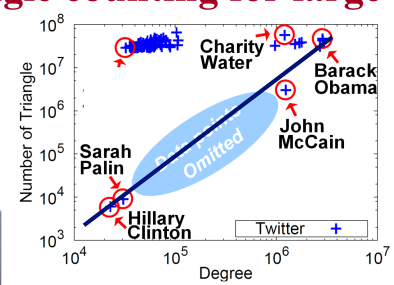
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)
[U Kang, Brendan Meeder, +, PAKDD'11]

Carnegie Mellon

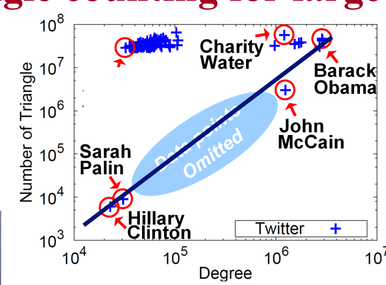
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)
[U Kang, Brendan Meeder, +, PAKDD'11]

Carnegie Mellon

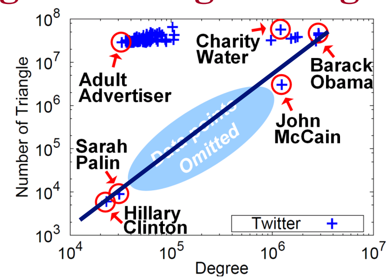
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)
[U Kang, Brendan Meeder, +, PAKDD'11]

CarnegieMellon

Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

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

CarnegieMellon

Any other 'laws'?

Yes!

CarnegieMellon

Any other 'laws'?

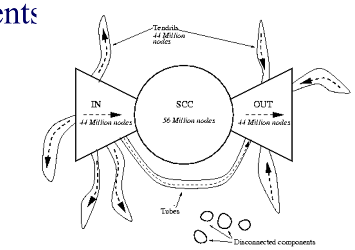
Yes!

- Small diameter (~ constant!) –
  – six degrees of separation / 'Kevin Bacon'
  – small worlds [Watts and Strogatz]

CarnegieMellon

Any other 'laws'?

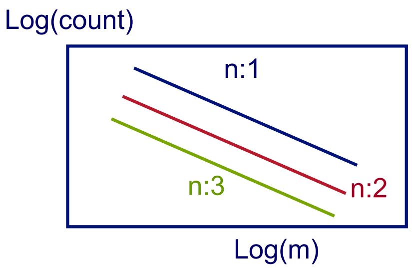
- Bow-tie, for the web [Kumar+ '99]
- IN, SCC, OUT, 'tendrils'
- disconnected components

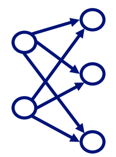


Carnegie Mellon

Any other 'laws'?

- power-laws in communities (bi-partite cores) [Kumar+, '99]



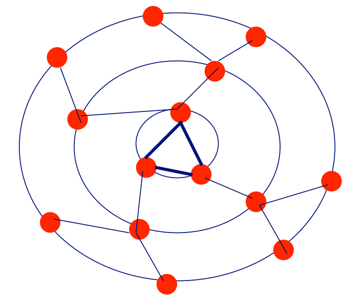


2:3 core
(m:n core)

Carnegie Mellon

Any other 'laws'?

- "Jellyfish" for Internet [Tauro+ '01]
- core: ~clique
- ~5 concentric layers
- many 1-degree nodes



Carnegie Mellon

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.

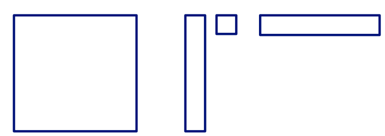
Useful for fraud detection!

Carnegie Mellon

EigenSpokes

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

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



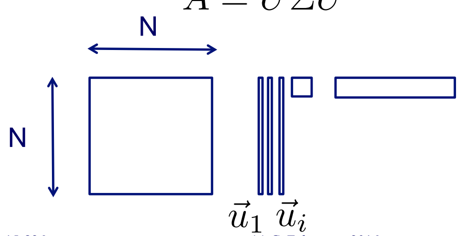
Carnegie Mellon

details

EigenSpokes

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

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



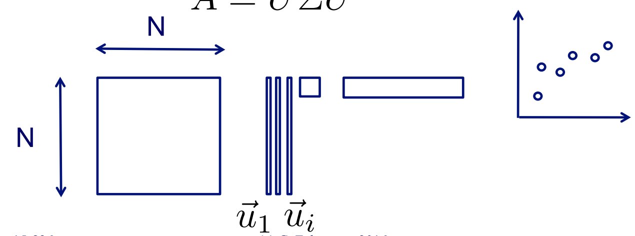
Carnegie Mellon

details

EigenSpokes

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

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



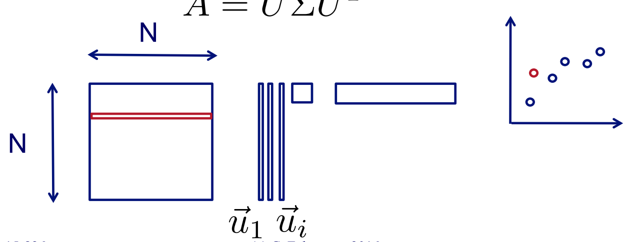
Carnegie Mellon

details

EigenSpokes

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

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



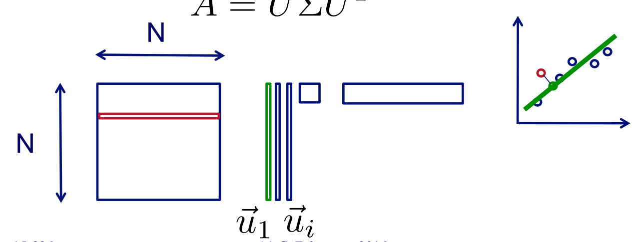
Carnegie Mellon

details

EigenSpokes

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

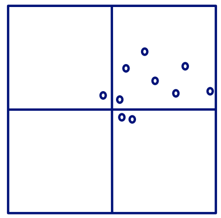
$$A = U \Sigma U^T$$



Carnegie Mellon

EigenSpokes

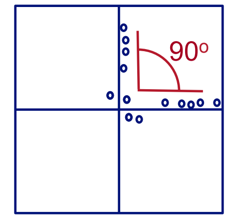
- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A few scattered ~randomly



Carnegie Mellon

EigenSpokes

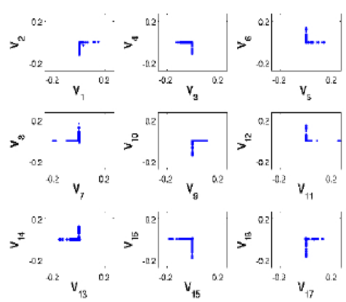
- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A few scattered ~randomly



Carnegie Mellon

EigenSpokes - pervasiveness

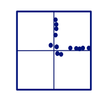
- Present in mobile social graph
  - across time and space
- Patent citation graph

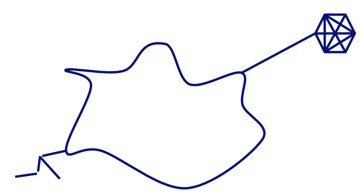


Carnegie Mellon

EigenSpokes - explanation

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

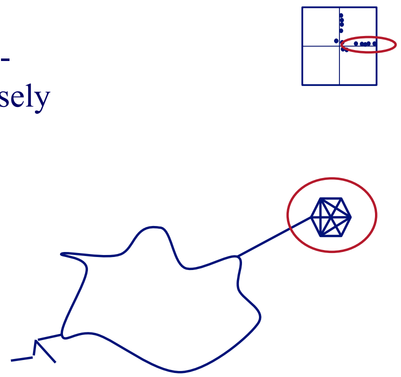




Carnegie Mellon

EigenSpokes - explanation

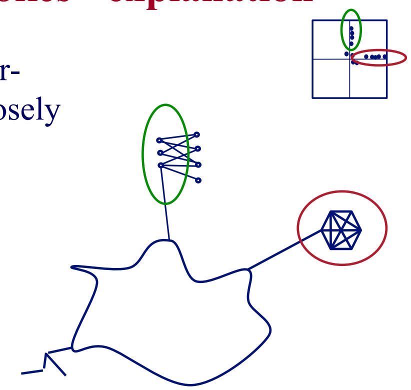
Near-cliques, or near-bipartite-cores, loosely connected



Carnegie Mellon

EigenSpokes - explanation

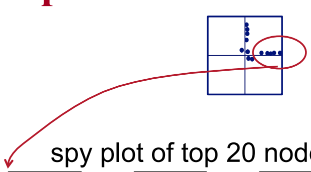
Near-cliques, or near-bipartite-cores, loosely connected



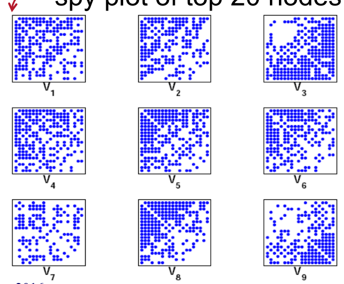
Carnegie Mellon

EigenSpokes - explanation

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



spy plot of top 20 nodes



So what?

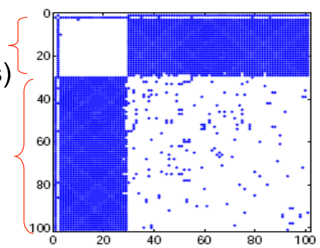
- Extract nodes with high *scores*
- high connectivity
- Good "communities"

Carnegie Mellon

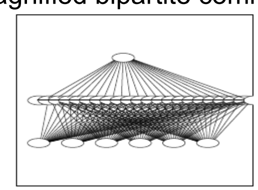
Bipartite Communities!

patents from same inventor(s)

`cut-and-paste' bibliography!



magnified bipartite community



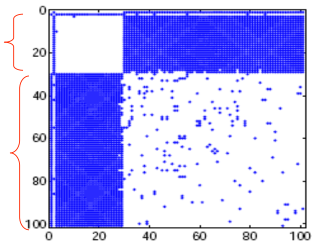
Useful for fraud detection!

Carnegie Mellon

Bipartite Communities!

IP – port scanners

victims

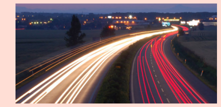


Useful for fraud detection!

15-826 (c) C. Faloutsos, 2016 69

Carnegie Mellon

Outline



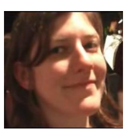
- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - Triangles
  - Weighted graphs
  - ➡ Time evolving graphs
- Problem#2: Scalability
- Conclusions

15-826 (c) C. Faloutsos, 2016 70

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

15-826 (c) C. Faloutsos, 2016 71

Carnegie Mellon

Observation W.1: Fortification

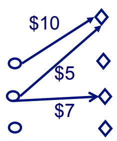
Q: How do the weights of nodes relate to degree?

15-826 (c) C. Faloutsos, 2016 72

CarnegieMellon

Observation W.1: Fortification

More donors, more $?



'Reagan'

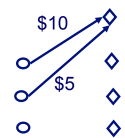
'Clinton'

CarnegieMellon

Observation W.1: fortification: Snapshot Power Law

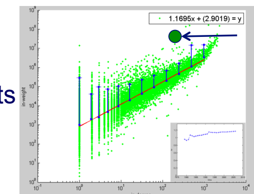
- Weight: super-linear on in-degree
- exponent 'iw' : 1.01 < iw < 1.26

More donors, even more $



In-weights ($)

Orgs-Candidates

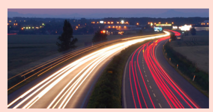


e.g. John Kerry, $10M received, from 1K donors

Edges (# donors)

CarnegieMellon

Outline



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

CarnegieMellon

Problem: Time evolution

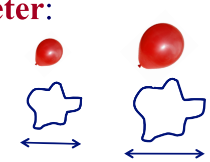
- with Jure Leskovec (CMU -> Stanford)

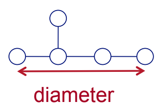
- and Jon Kleinberg (Cornell – sabb. @ CMU)

Carnegie Mellon

T.1 Evolution of the Diameter

- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - [diameter ~ O($N^{1/3}$)]
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?

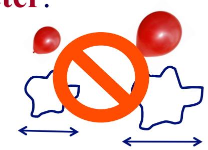




Carnegie Mellon

T.1 Evolution of the Diameter

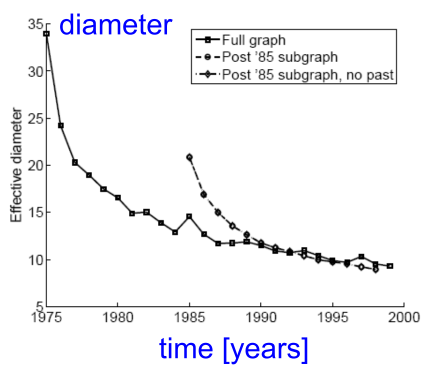
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - [diameter ~ O($N^{1/3}$)]
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?
- Diameter **shrinks** over time



Carnegie Mellon

T.1 Diameter – "Patents"

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



Carnegie Mellon

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)

Carnegie Mellon

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

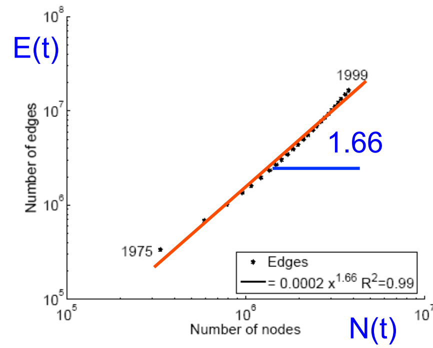
  E(t+1) =? 2 * E(t)
- A: over-doubled!
  - But obeying the ``Densification Power Law''

15-826 (c) C. Faloutsos, 2016 81

Carnegie Mellon

T.2 Densification – Patent Citations

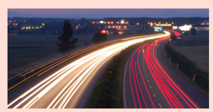
- Citations among patents granted
- @1999
  - 2.9 M nodes
  - 16.5 M edges
- Each year is a datapoint



15-826 (c) C. Faloutsos, 2016 82

Carnegie Mellon

Outline



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

15-826 (c) C. Faloutsos, 2016 83

Carnegie Mellon

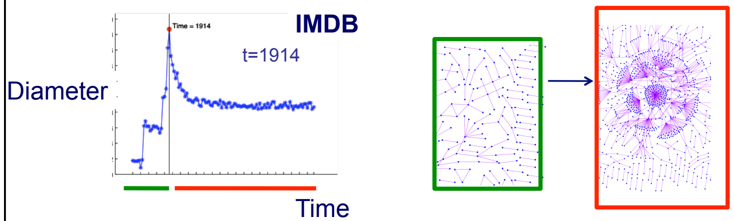
More on Time-evolving graphs

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

15-826 (c) C. Faloutsos, 2016 84

[Gelling Point]

- Most real graphs display a gelling point
- After gelling point, they exhibit typical behavior. This is marked by a spike in diameter.



Observation T.3: NLCC behavior

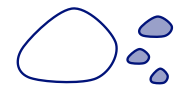
Q: How do NLCC's emerge and join with the GCC?

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

– Do they continue to grow in size?

– or do they shrink?

– or stabilize?



Observation T.3: NLCC behavior

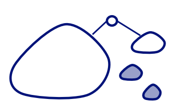
Q: How do NLCC's emerge and join with the GCC?

(``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?

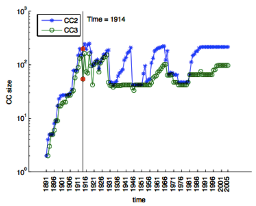
YES – or do they shrink?

YES – or stabilize?

CarnegieMellon

Observation T.3: NLCC behavior

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



CarnegieMellon

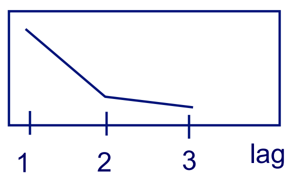
Timing for Blogs

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

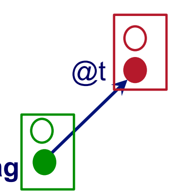
[SDM' 07]

CarnegieMellon

T.4 : popularity over time

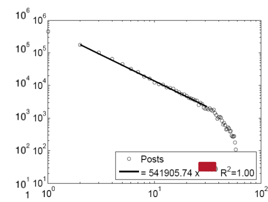


Post popularity drops-off – exponentially?



CarnegieMellon

T.4 : popularity over time

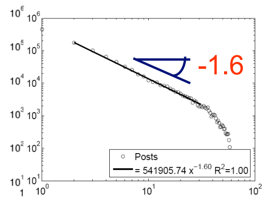


Post popularity drops-off – exponentially?
POWER LAW!
Exponent?

CarnegieMellon

T.4 : popularity over time

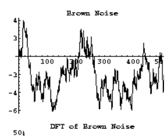
in links (**log**)



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



CarnegieMellon

-1.5 slope

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

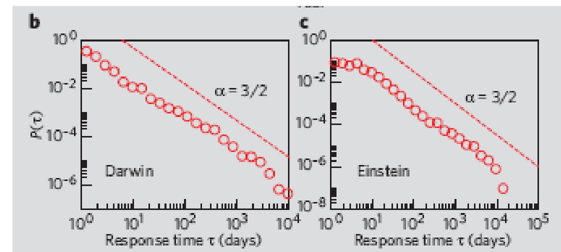


Figure 1 | The correspondence patterns of Darwin and Einstein.

CarnegieMellon

T.5: duration of phonecalls

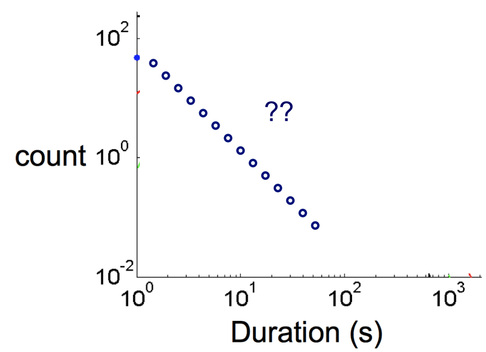
Surprising Patterns for the Call Duration Distribution of Mobile Phone Users

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

PKDD 2010

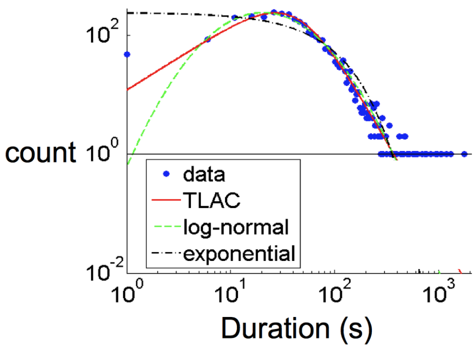
CarnegieMellon

Probably, power law (?)



Carnegie Mellon

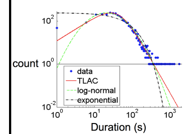
No Power Law!



Carnegie Mellon

'TLaC: Lazy Contractor'

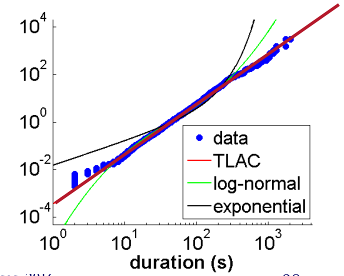
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

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

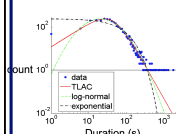
== power law



Carnegie Mellon

Log-logistic distribution

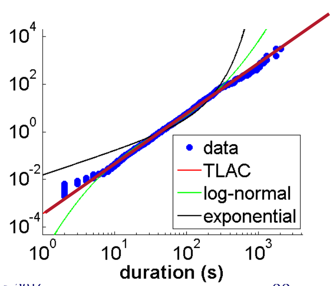
- CDF(t)/(1- CDF(t)) == OR(t)
- For log-logistic: log[OR(t)] = β + ρ*log(t)



Odds ratio=

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

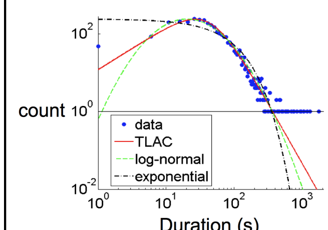
== power law

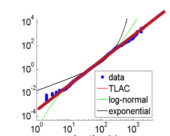


Carnegie Mellon

Log-logistic distribution

- CDF(t)/(1- CDF(t)) == OR(t)
- For log-logistic: log[OR(t)] = β + ρ*log(t)





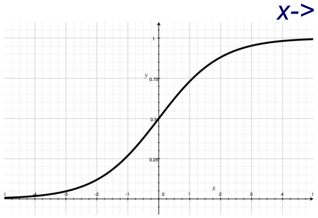
- PDF looks like hyperbola;
- and, if clipped, like power-law

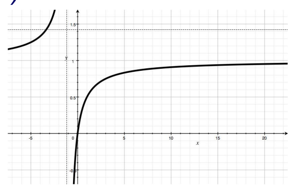
CarnegieMellon

Log-logistic distribution

- Logistic distribution: CDF -> sigmoid
- LOG-Logistic distribution:

x-> ln(x)





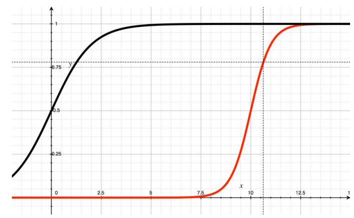
CDF(x) = 1/(1+exp(-x))

CDF(x) = 1/(1+1/x)

CarnegieMellon

Log-logistic distribution

- Logistic distribution: CDF -> sigmoid
- LOG-Logistic distribution:



CDF(x) = 1/(1+exp(-(x-m)/s))

CDF(x) = 1/(1+exp(-(ln(x)-m)/s))

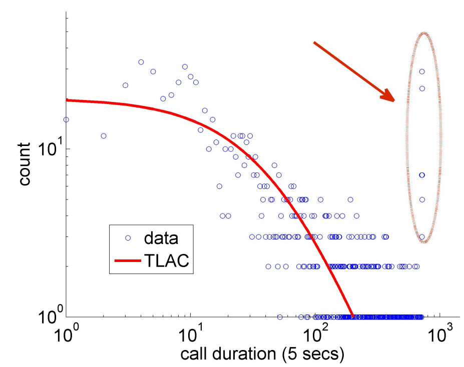
CarnegieMellon

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)

CarnegieMellon

Outliers:



Carnegie Mellon

Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- ➡ Problem#2: Scalability -PEGASUS
- Conclusions

15-826 (c) C. Faloutsos, 2016 105

Carnegie Mellon

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/

hadoop

15-826 (c) C. Faloutsos, 2016 106

Carnegie Mellon

Outline – 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**	

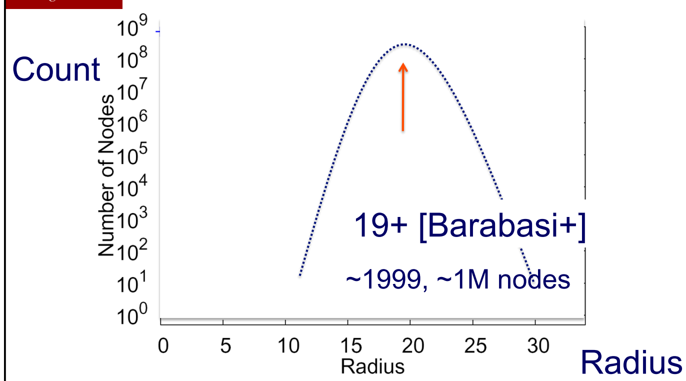
15-826 (c) C. Faloutsos, 2016 107

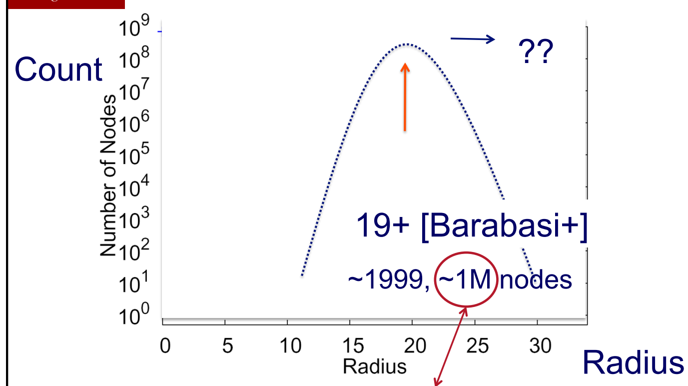
Carnegie Mellon

HADI for diameter estimation

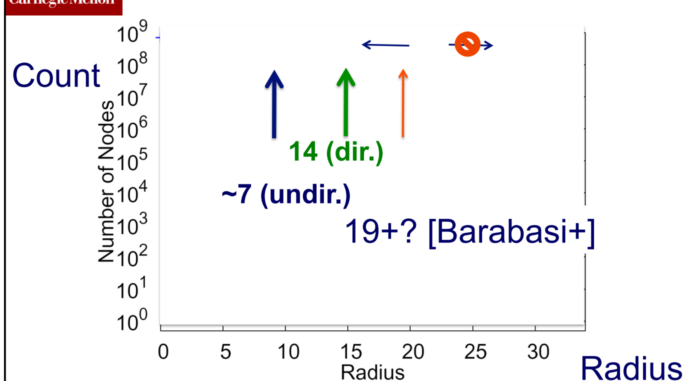
- *Radius Plots for Mining Tera-byte Scale Graphs* **U Kang**, Charalampos Tsourakakis, Ana Paula Appel, Christos Faloutsos, Jure Leskovec, SDM'10
- Naively: diameter needs **O(N**2)** space and up to O(N**3) time – **prohibitive** (N~1B)
- Our HADI: linear on E (~10B)
  - Near-linear scalability wrt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

15-826 (c) C. Faloutsos, 2016 108

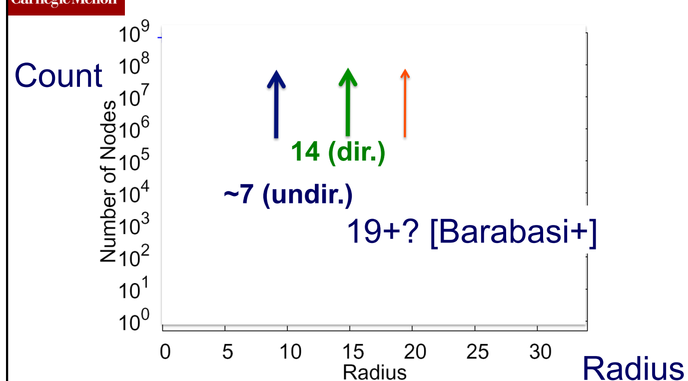




YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- Largest publicly available graph ever studied.

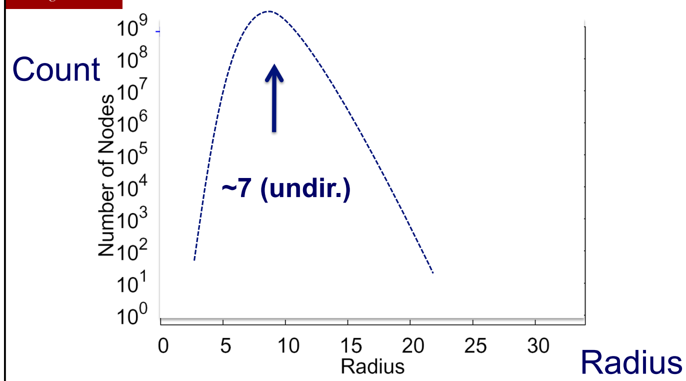


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- Largest publicly available graph ever studied.



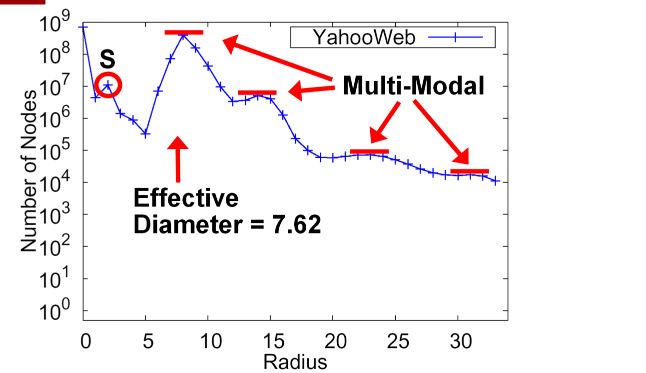
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- 7 degrees of separation (!)
- Diameter: shrunk

Carnegie Mellon



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

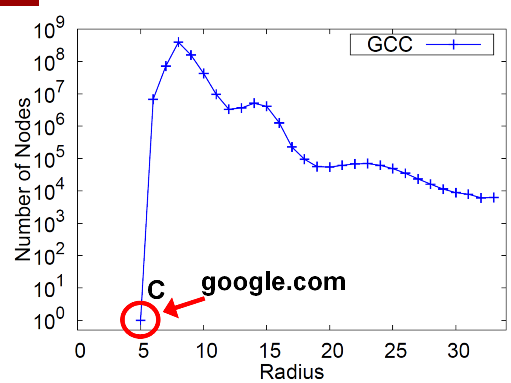
Carnegie Mellon



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

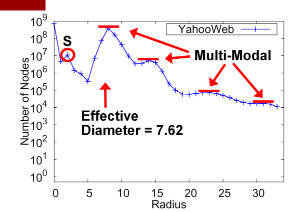
- effective diameter: surprisingly small.
- Multi-modality (?!)

Carnegie Mellon



Radius Plot of **GCC** of YahooWeb.

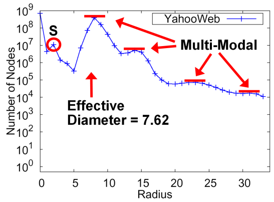
Carnegie Mellon



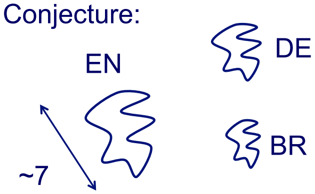
YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)

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

Carnegie Mellon

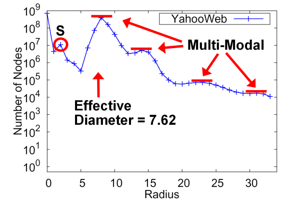


Conjecture:

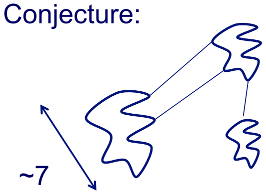


YahooWeb graph (120Gb, 1.4B nodes, 6.6 B edges)
- effective diameter: surprisingly small.
- Multi-modality: probably mixture of cores .

Carnegie Mellon



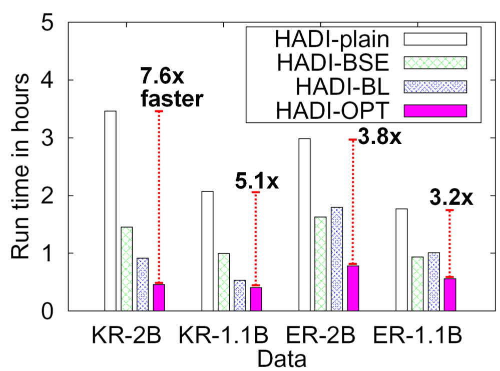
Conjecture:



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

Outline – 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).

Carnegie Mellon

Generalized Iterated Matrix Vector Multiplication (GIMV)

details

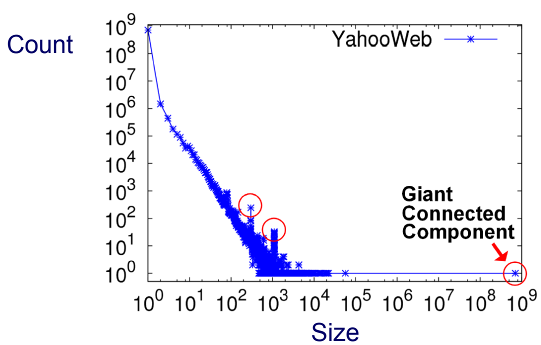
- PageRank
- proximity (RWR)
- Diameter
- Connected components
- (eigenvectors,
- Belief Prop.
- …)

Matrix – vector Multiplication (iterated)

Carnegie Mellon

Example: GIM-V At Work

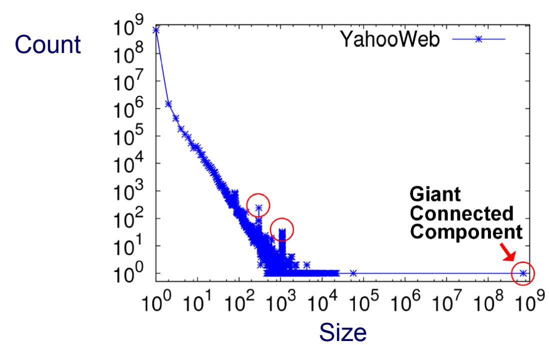
- Connected Components – 4 observations:



Carnegie Mellon

Example: GIM-V At Work

- Connected Components

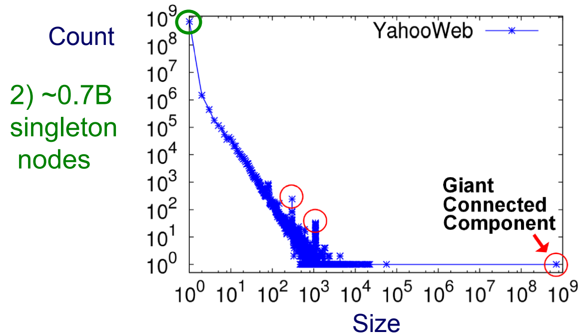


1) 10K x larger than next

Carnegie Mellon

Example: GIM-V At Work

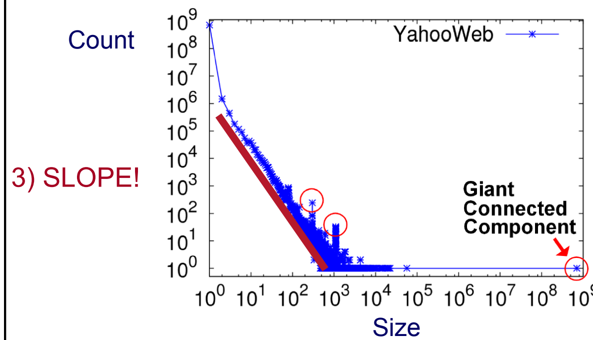
- Connected Components



Carnegie Mellon

Example: GIM-V At Work

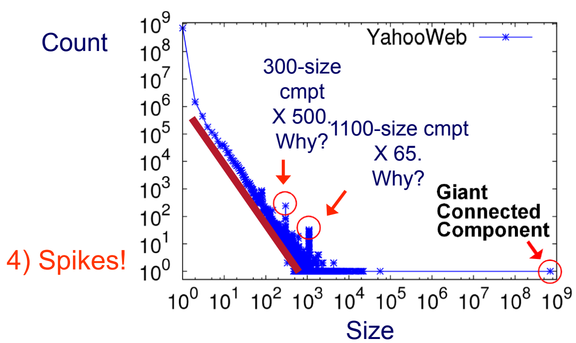
- Connected Components



Carnegie Mellon

Example: GIM-V At Work

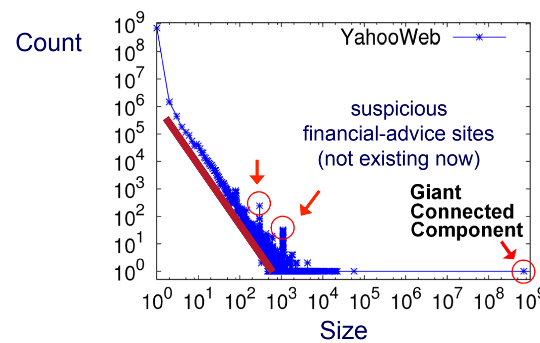
- Connected Components



Carnegie Mellon

Example: GIM-V At Work

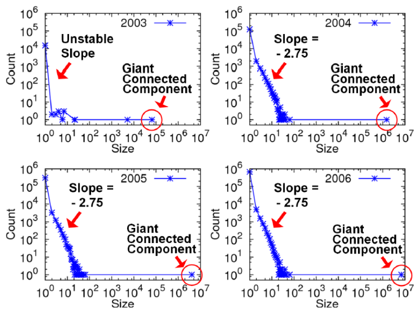
- Connected Components



Carnegie Mellon

GIM-V At Work

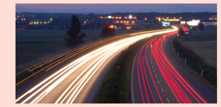
- Connected Components over Time
- **LinkedIn: 7.5M nodes and 58M edges**



Stable tail slope after the gelling point

Carnegie Mellon

Outline



- Introduction – Motivation
- Problem#1: Patterns in graphs
- DELETE
- Problem#2: Scalability
- ➡ Conclusions

Carnegie Mellon

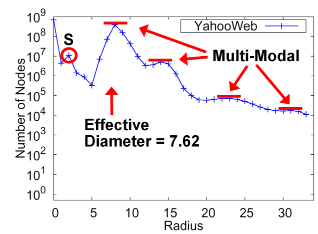
OVERALL CONCLUSIONS – low level:

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

Carnegie Mellon

OVERALL CONCLUSIONS – high level

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



CarnegieMellon

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)

15-826 (c) C. Faloutsos, 2016 133

CarnegieMellon

References

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

15-826 (c) C. Faloutsos, 2016 134

CarnegieMellon

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

15-826 (c) C. Faloutsos, 2016 135

CarnegieMellon

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

15-826 (c) C. Faloutsos, 2016 136

Carnegie Mellon

References

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

15-826 (c) C. Faloutsos, 2016 137

Carnegie Mellon

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

15-826 (c) C. Faloutsos, 2016 138

Carnegie Mellon

References

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

15-826 (c) C. Faloutsos, 2016 139

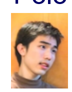
Carnegie Mellon

(Project info)

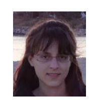
www.cs.cmu.edu/~pegasus

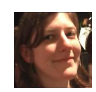
Project Pegasus


Akoglu, Leman

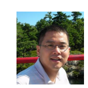

Chau, Polo


Kang, U


Koutra, Danae


McGlohon, Mary


Prakash, Aditya


Tong, Hanghang

15-826 (c) C. Faloutsos, 2016 140