Carnegie Mellon

Talk 1: Graph Mining – patterns and generators

Christos Faloutsos

CMU

Our goal:

Open source system for mining huge graphs:

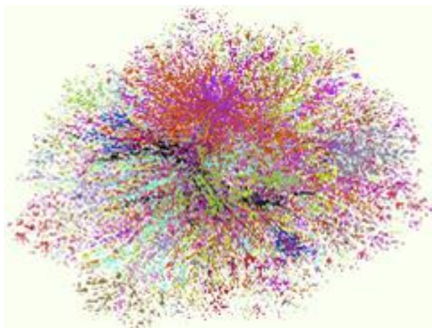
PEGASUS project (PEta GrAph mining System)

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

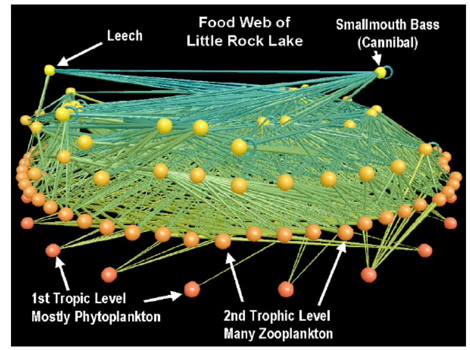
Outline

- Introduction – Motivation
- Talk#1: Patterns in graphs; generators
- Talk#2: Tools (Ranking, proximity)
- Talk#3: Tools (Tensors, scalability)
- Conclusions

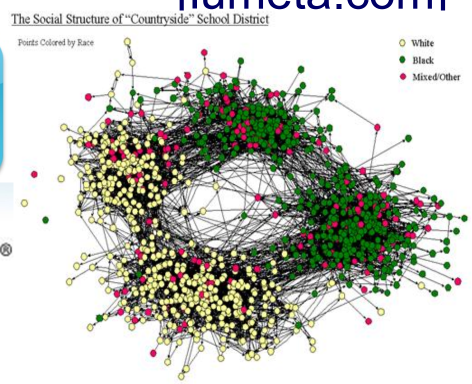
Graphs - why should we care?



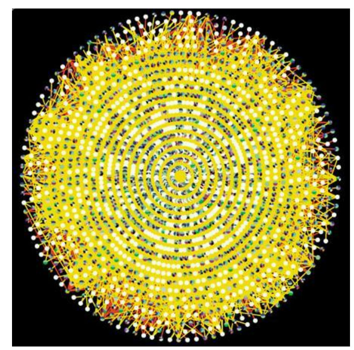
Internet Map
[lumeta.com]



Food Web
[Martinez '91]



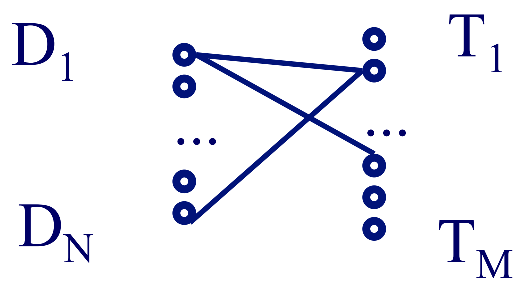
Friendship Network
[Moody '01]



Protein Interactions
[genomebiology.com]

Graphs - why should we care?

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



- web: hyper-text graph

- ... and more:

Graphs - why should we care?

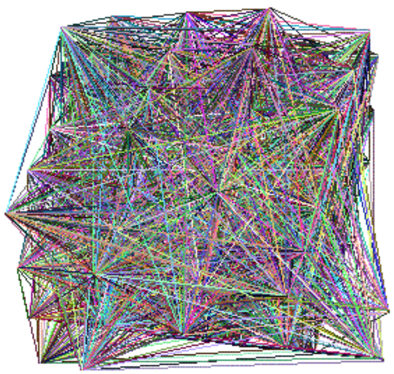
- network of companies & board-of-directors members
- ‘viral’ marketing
- web-log (‘blog’) news propagation
- computer network security: email/IP traffic and anomaly detection
- ....

Carnegie Mellon

Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

Network and graph mining

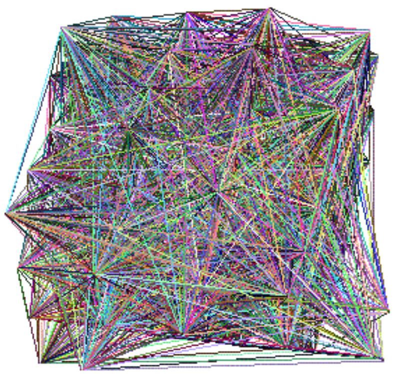


- How does the Internet look like?
- How does FaceBook look like?

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

Carnegie Mellon

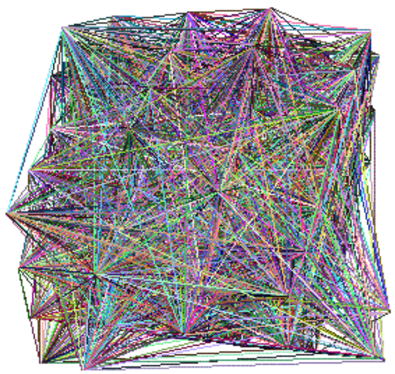
Network and graph mining



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

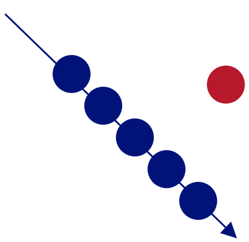
- What is 'normal'/'abnormal'?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**

Network and graph mining



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

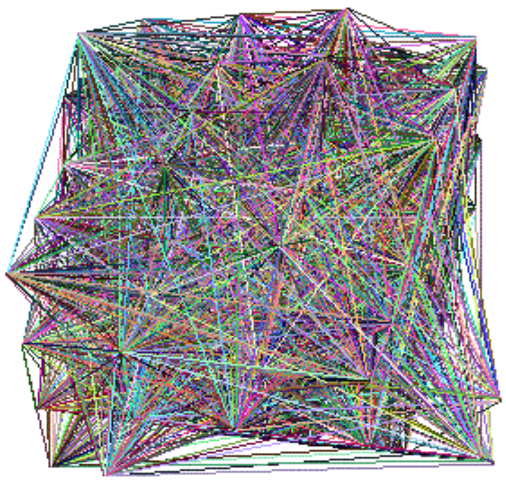
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?
  - To spot **anomalies** (rarities), we have to discover **patterns**
  - **Large** datasets reveal patterns/anomalies that may be invisible otherwise…



Carnegie Mellon

Topology

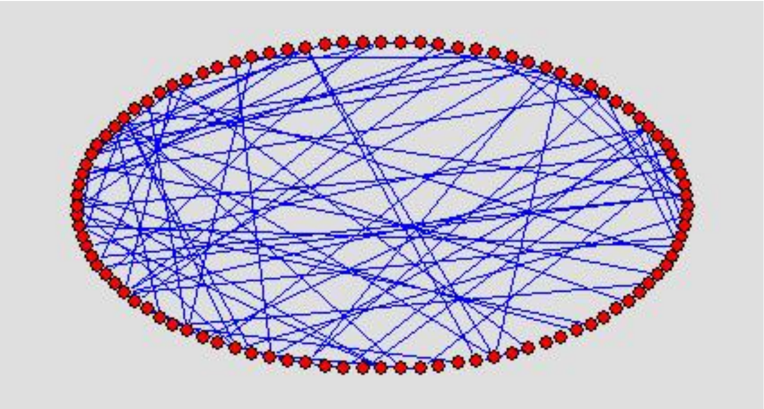
How does the Internet look like? Any rules?

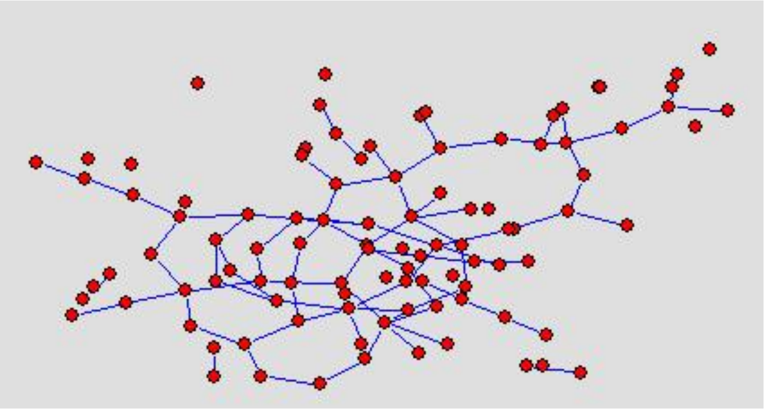


(Looks random – right?)

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?

Laws and patterns

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

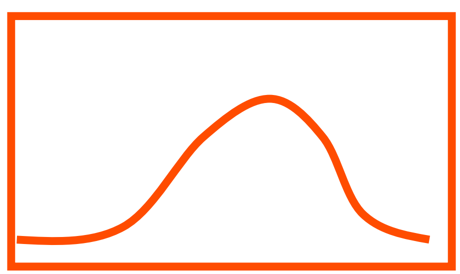
- So, let's look at the data

Laws – degree distributions

- Q: avg degree is ~2 - what is the most probable degree?

count

??



2

degree

Carnegie Mellon

Laws – degree distributions

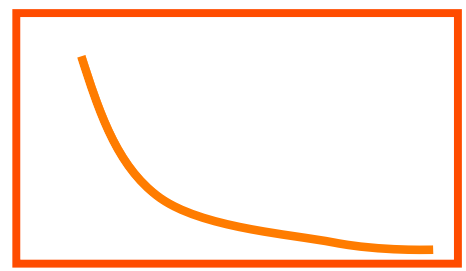
- Q: avg degree is ~2 - what is the most probable degree?

count

2

degree

count

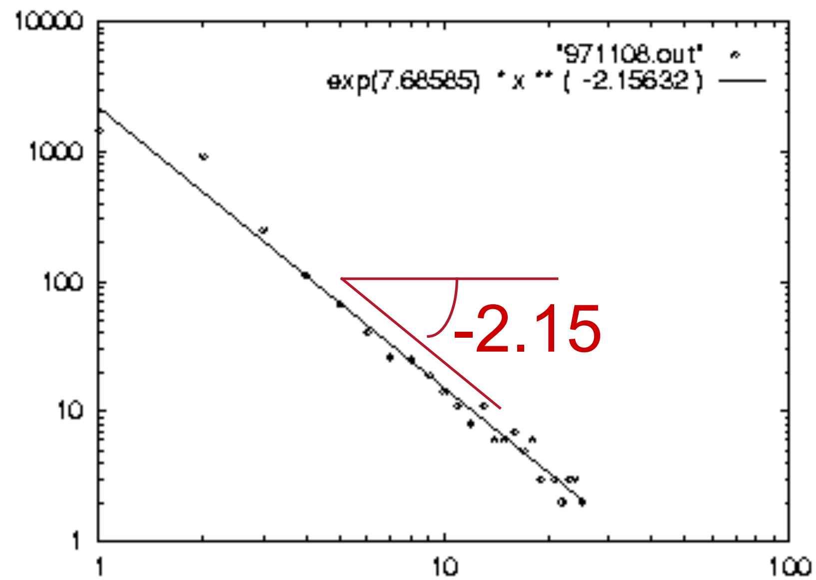


2

degree

Carnegie Mellon

Solution S1 .Power-law: outdegree *O*



Exponent = slope

O = -2.15

Nov'97

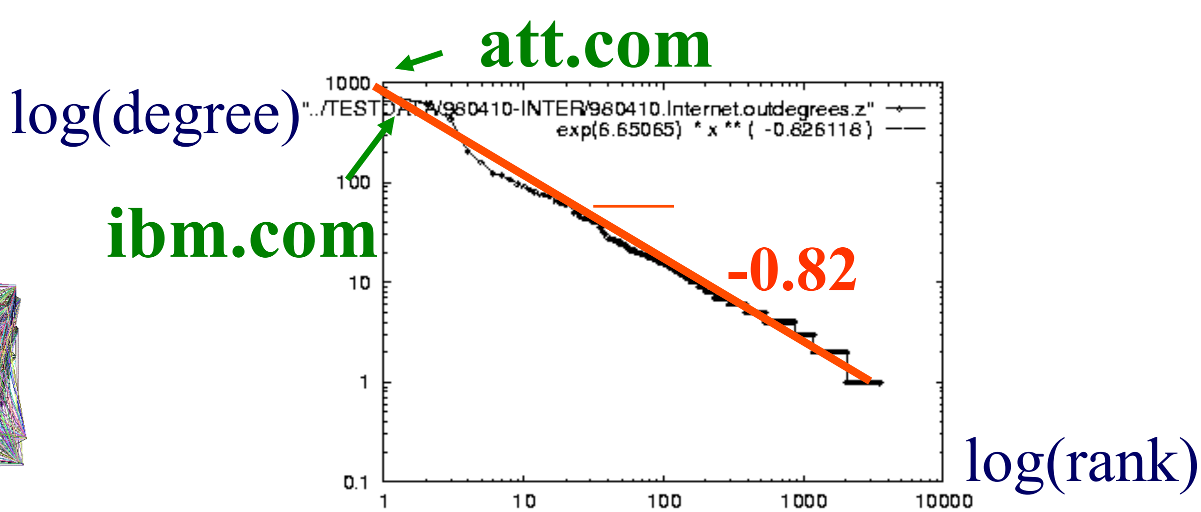
The plot is linear in log-log scale [FFF'99]

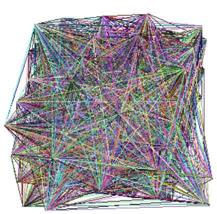
$$freq = degree^{(-2.15)}$$

Solution# S.1'

- Power law in the degree distribution [SIGCOMM99]

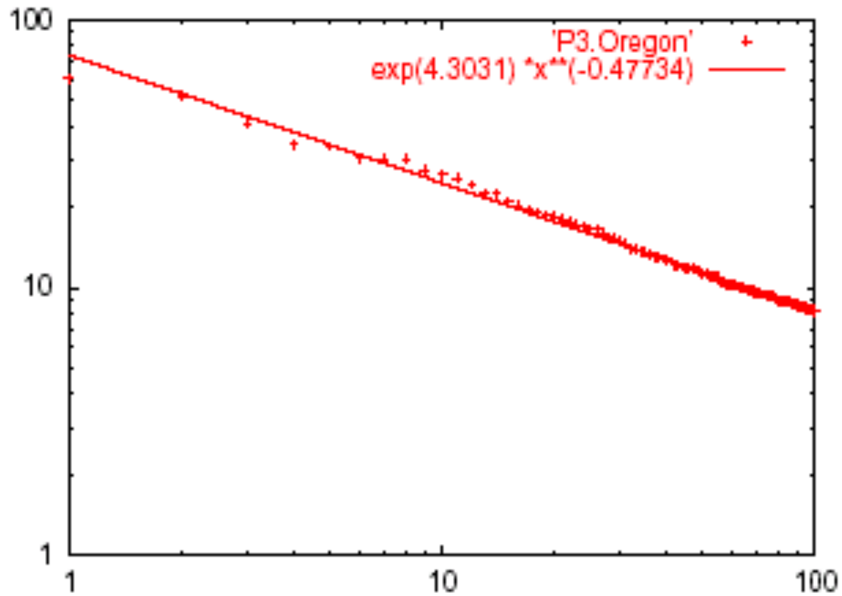
internet domains





Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

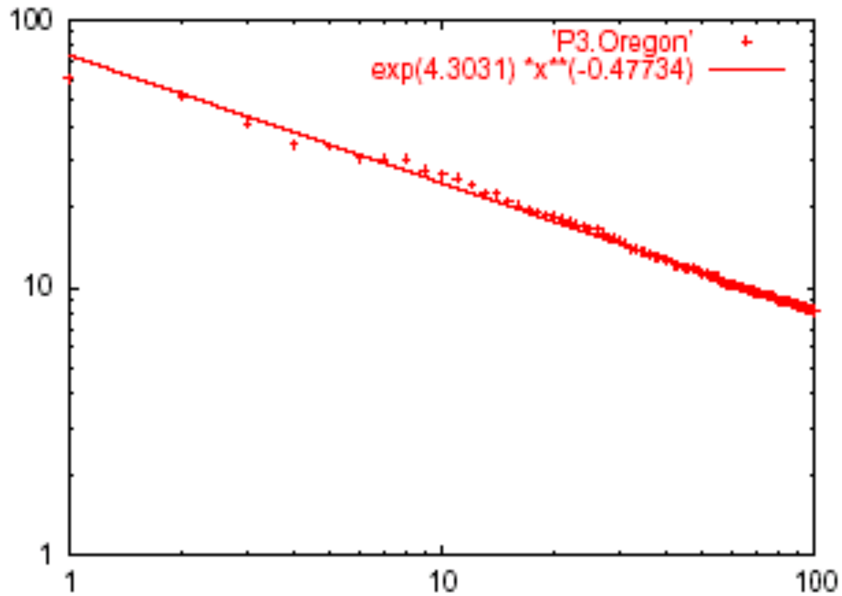
E = -0.48

May 2001

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

Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

E = -0.48

May 2001

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

But:

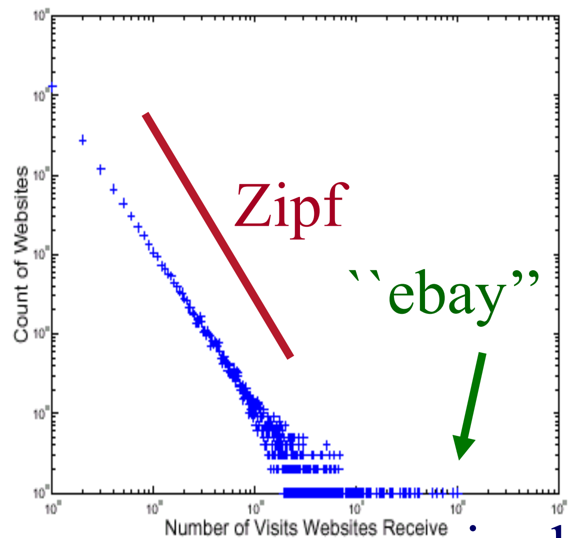
How about graphs from other domains?

More power laws:

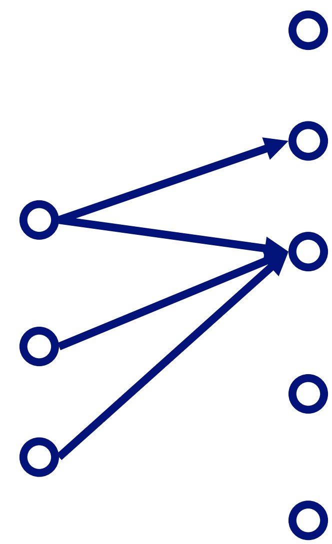
- web hit counts [w/ A. Montgomery]

Web Site Traffic

Count
(log scale)



in-degree (log scale)



users

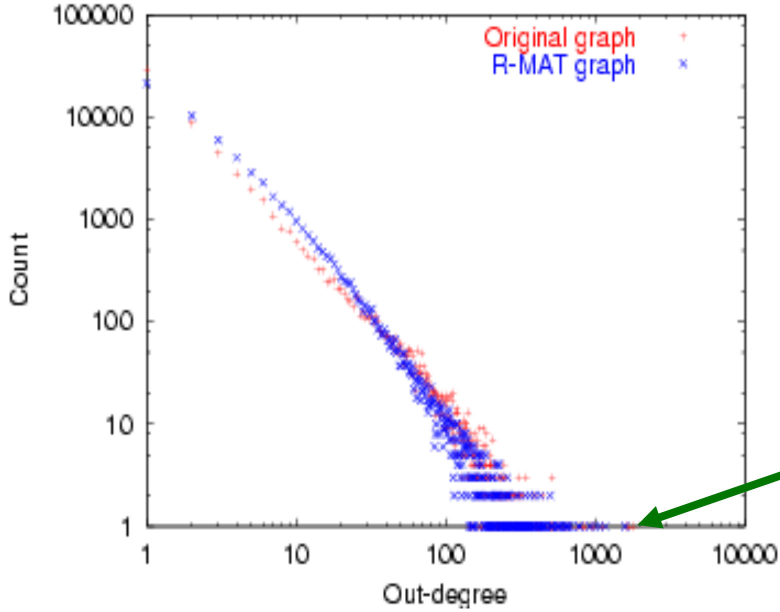
sites

Carnegie Mellon

epinions.com

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

count



trusts-2000-people user

(out) degree

And numerous more

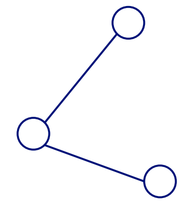
- # of sexual contacts
- Income [Pareto] –'80-20 distribution'
- Duration of downloads [Bestavros+]
- Duration of UNIX jobs ('mice and elephants')
- Size of files of a user
- …
- 'Black swans'

Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
    - Degree
    - Triangles
    - …
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

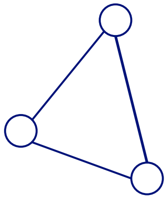
Carnegie Mellon

Solution# S.3: Triangle 'Laws'



- Real social networks have a lot of triangles

Solution# S.3: Triangle 'Laws'

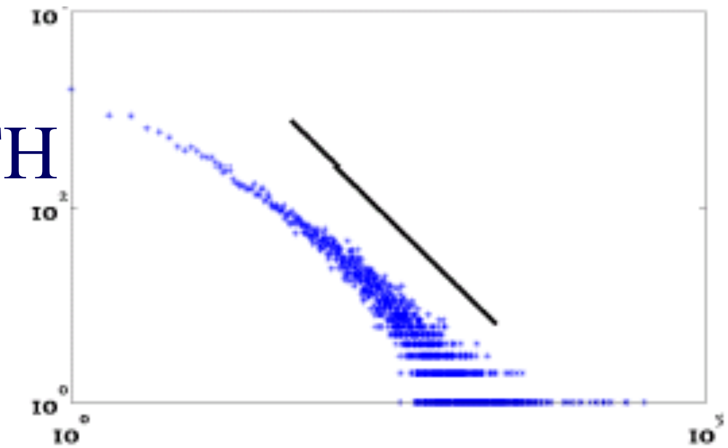


- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

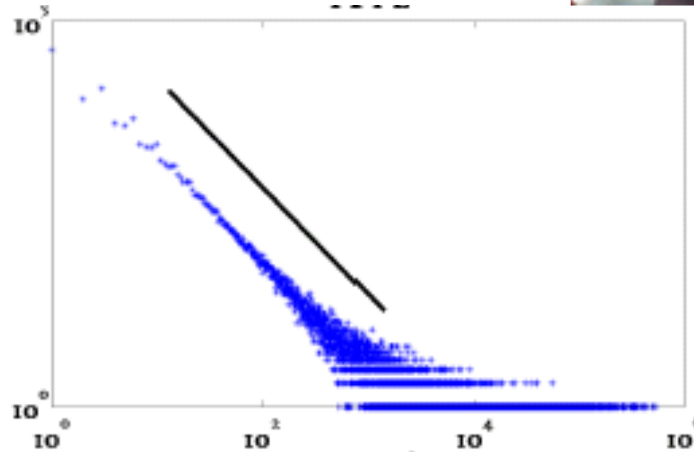
Carnegie Mellon

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

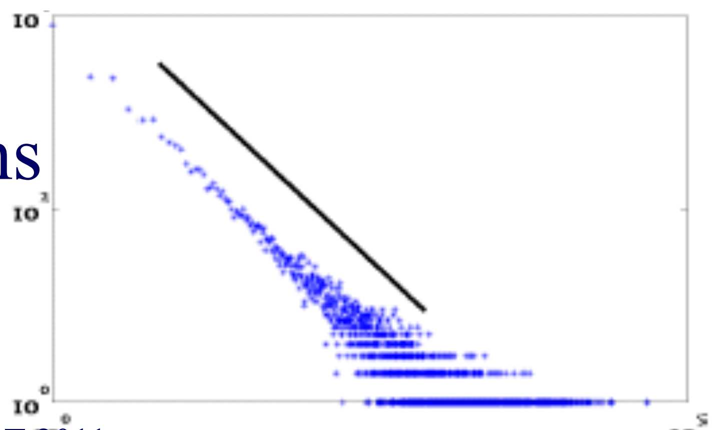
HEP-TH



ASN



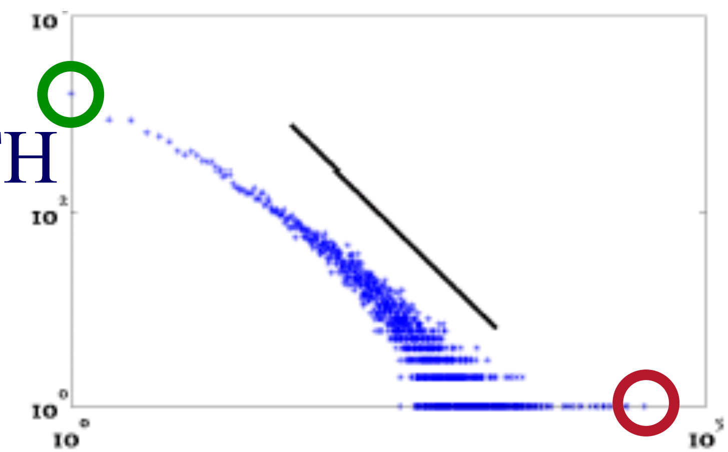
Epinions



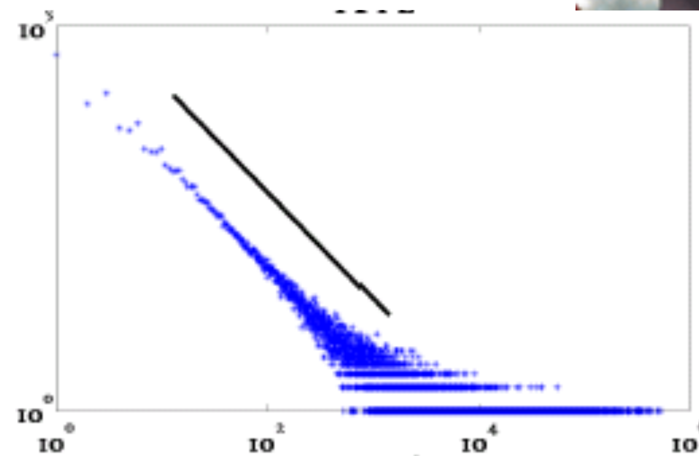
X-axis: # of Triangles
a node participates in
Y-axis: count of such nodes

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

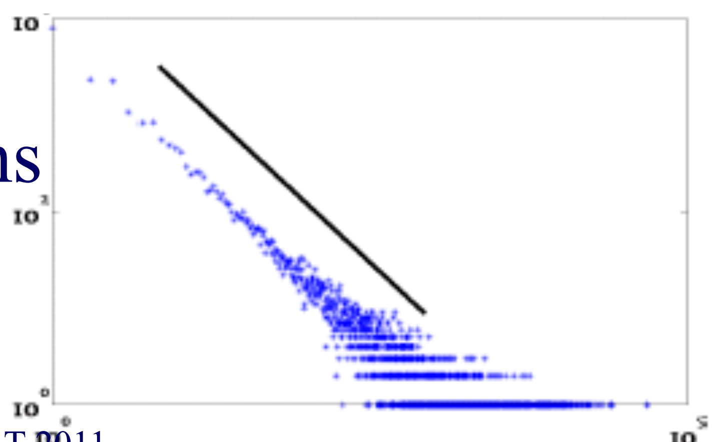
HEP-TH



ASN



Epinions



X-axis: # of Triangles
a node participates in
Y-axis: count of such nodes

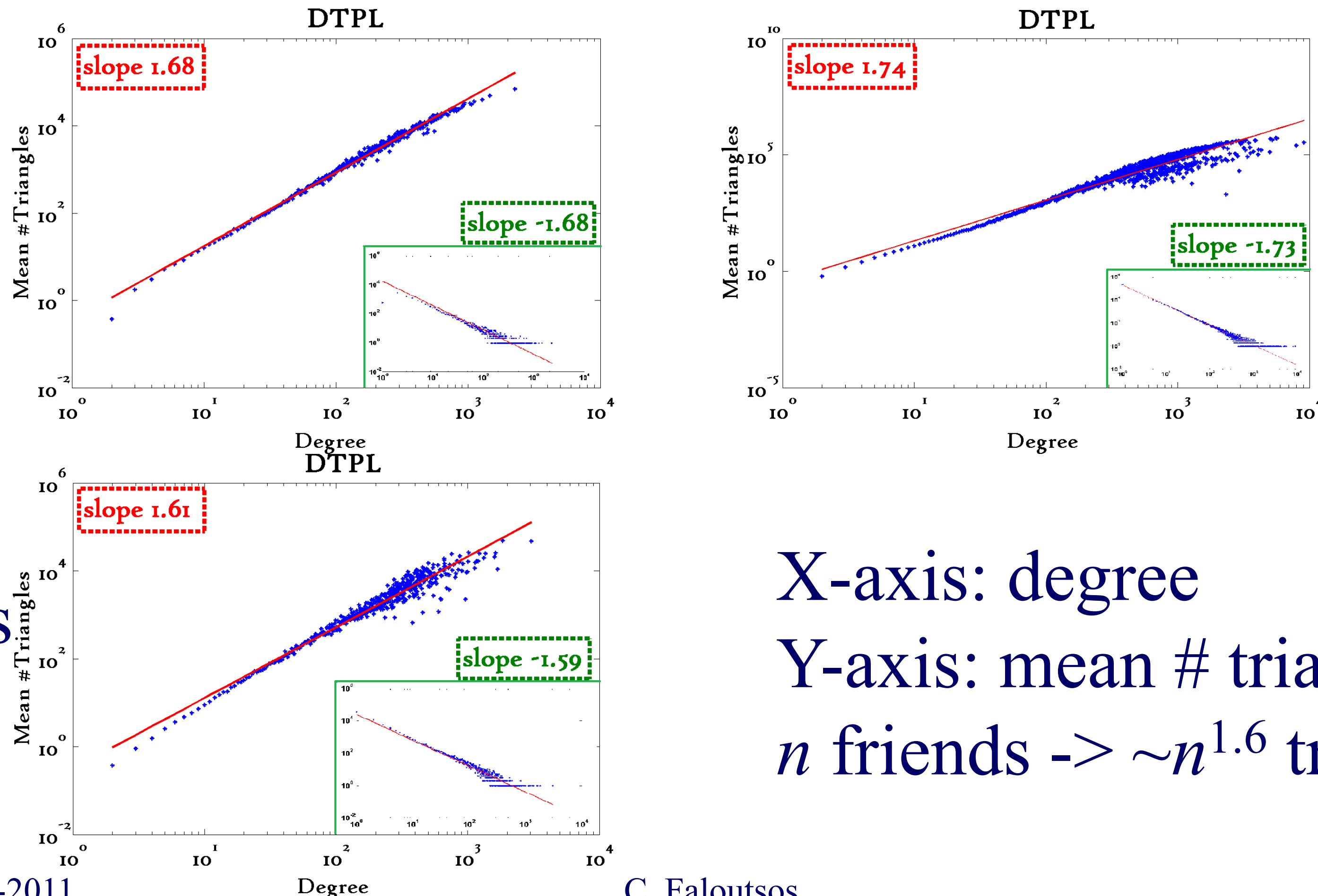
Carnegie Mellon

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

Reuters

SN

Epinions



X-axis: degree

Y-axis: mean # triangles

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

details

Triangle Law: Computations
[Tsourakakis ICDM 2008]

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

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?

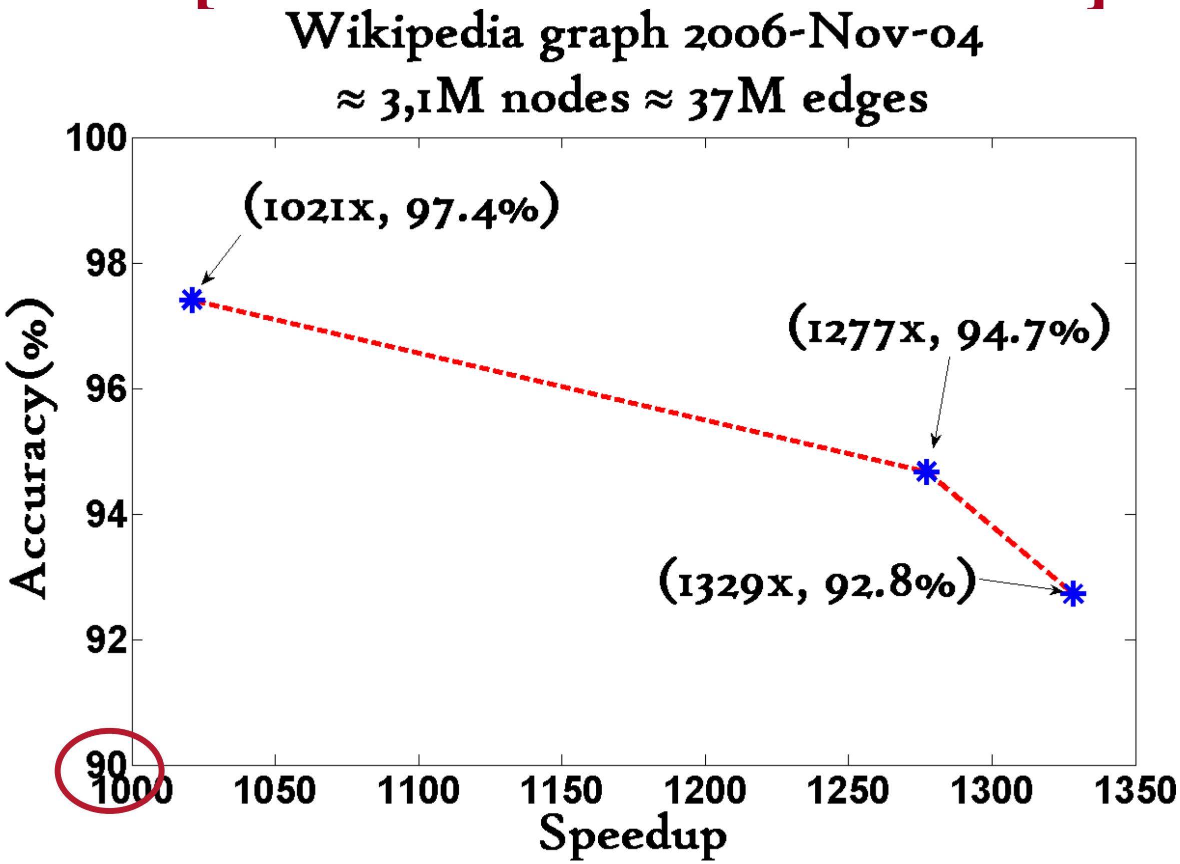
A: Yes!

#triangles = 1/6 Sum (λ_i^3)

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

details

Triangle Law: Computations
[Tsourakakis ICDM 2008]



1000x+ speed-up, >90% accuracy

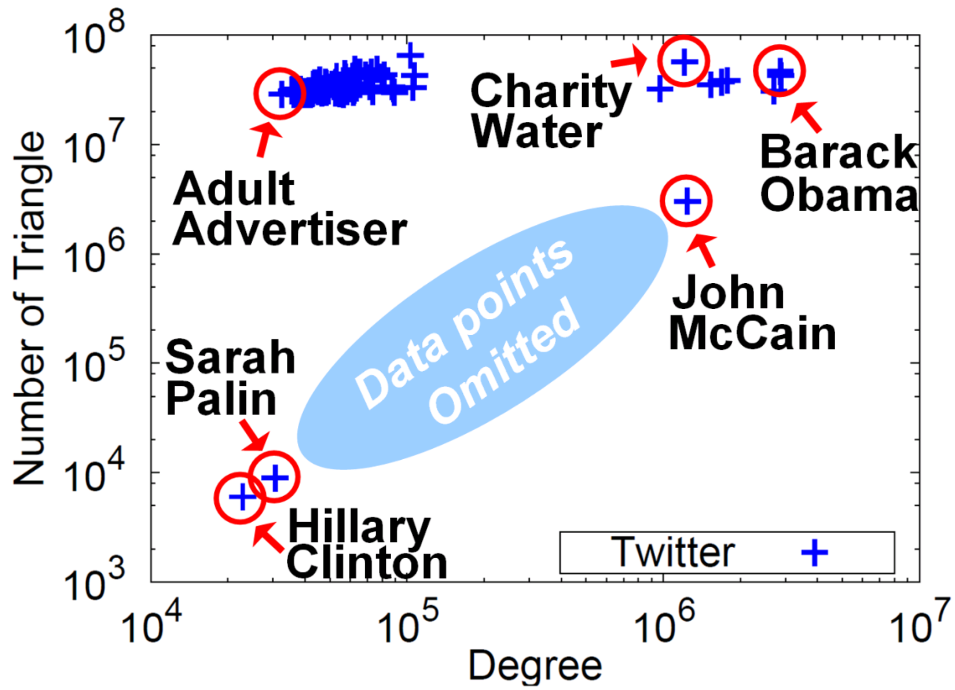
Triangle counting for large graphs?

Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

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

How about cliques?

Large Human Communication Networks

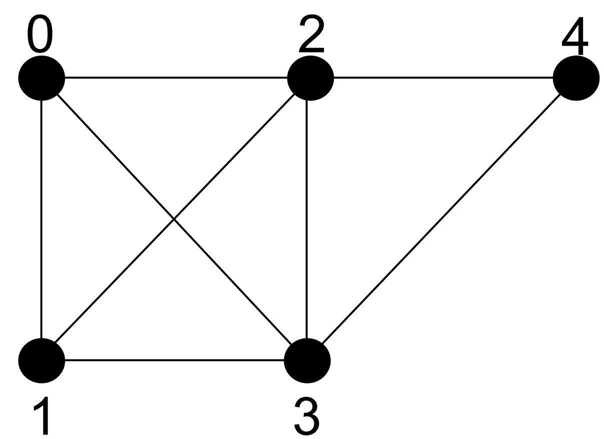
Patterns and a Utility-Driven Generator

Nan Du, Christos Faloutsos, Bai Wang, Leman Akoglu

KDD 2009

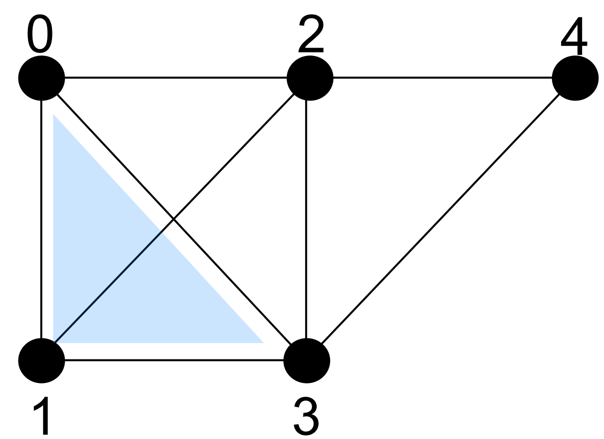
Cliques

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



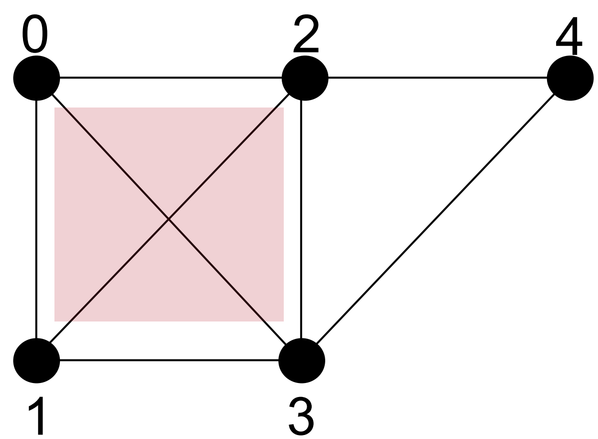
Clique

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



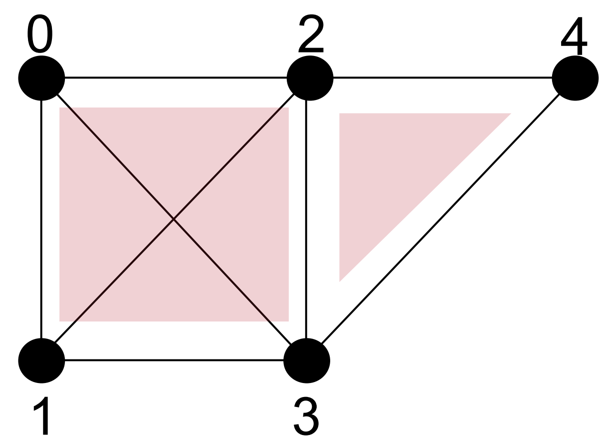
Clique

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



Clique

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



Carnegie Mellon

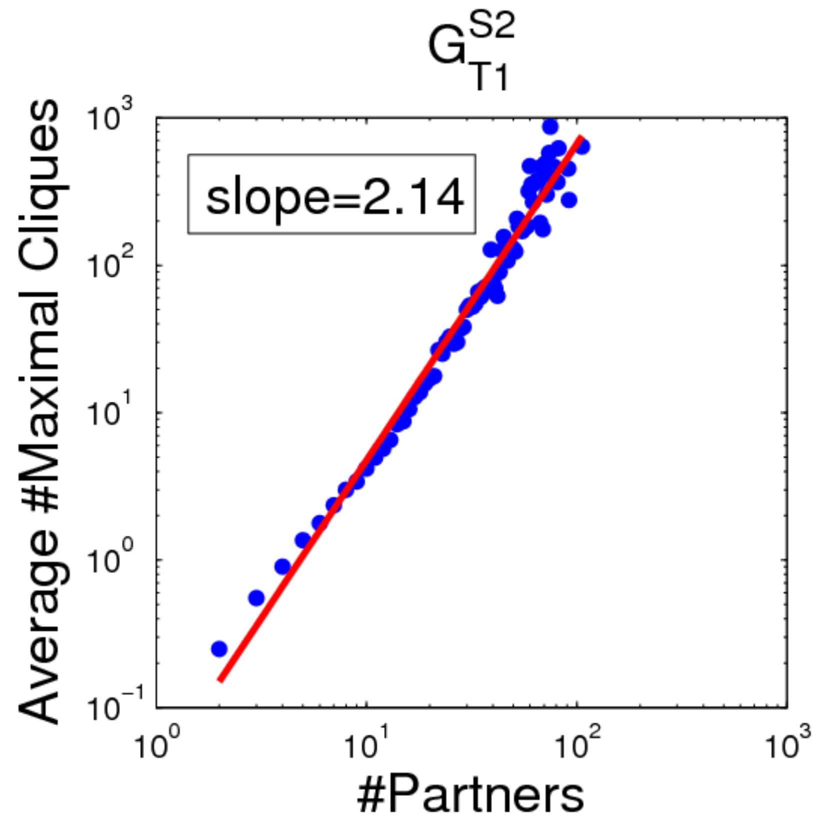
S5: Clique-Degree Power-Law

- Power law:

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

\# maximal cliques of node i

degree of node i

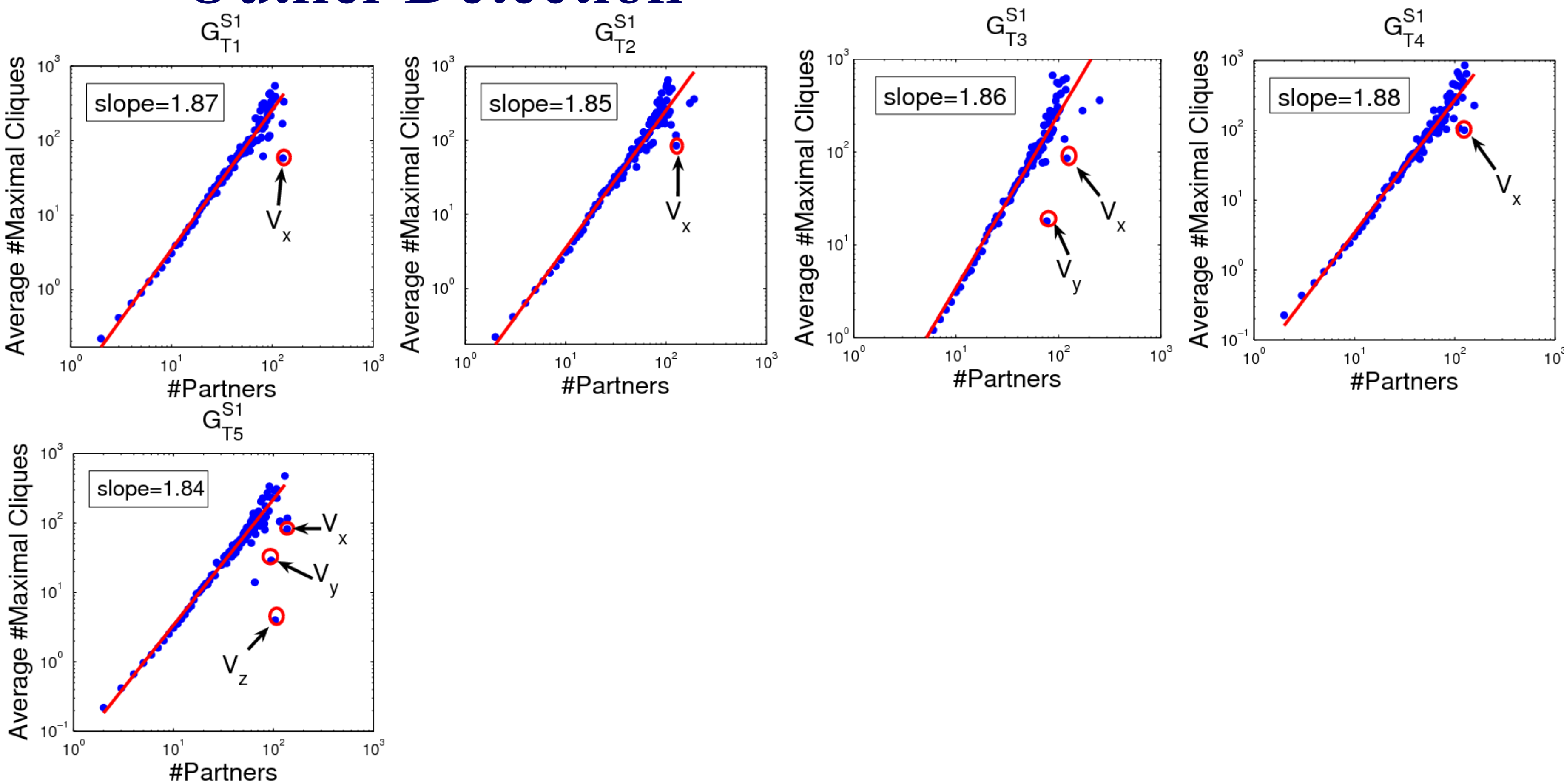


α is the power law exponent
$\alpha \in [1.8, 2.2]$ for S1~S3

More friends, even more social circles !

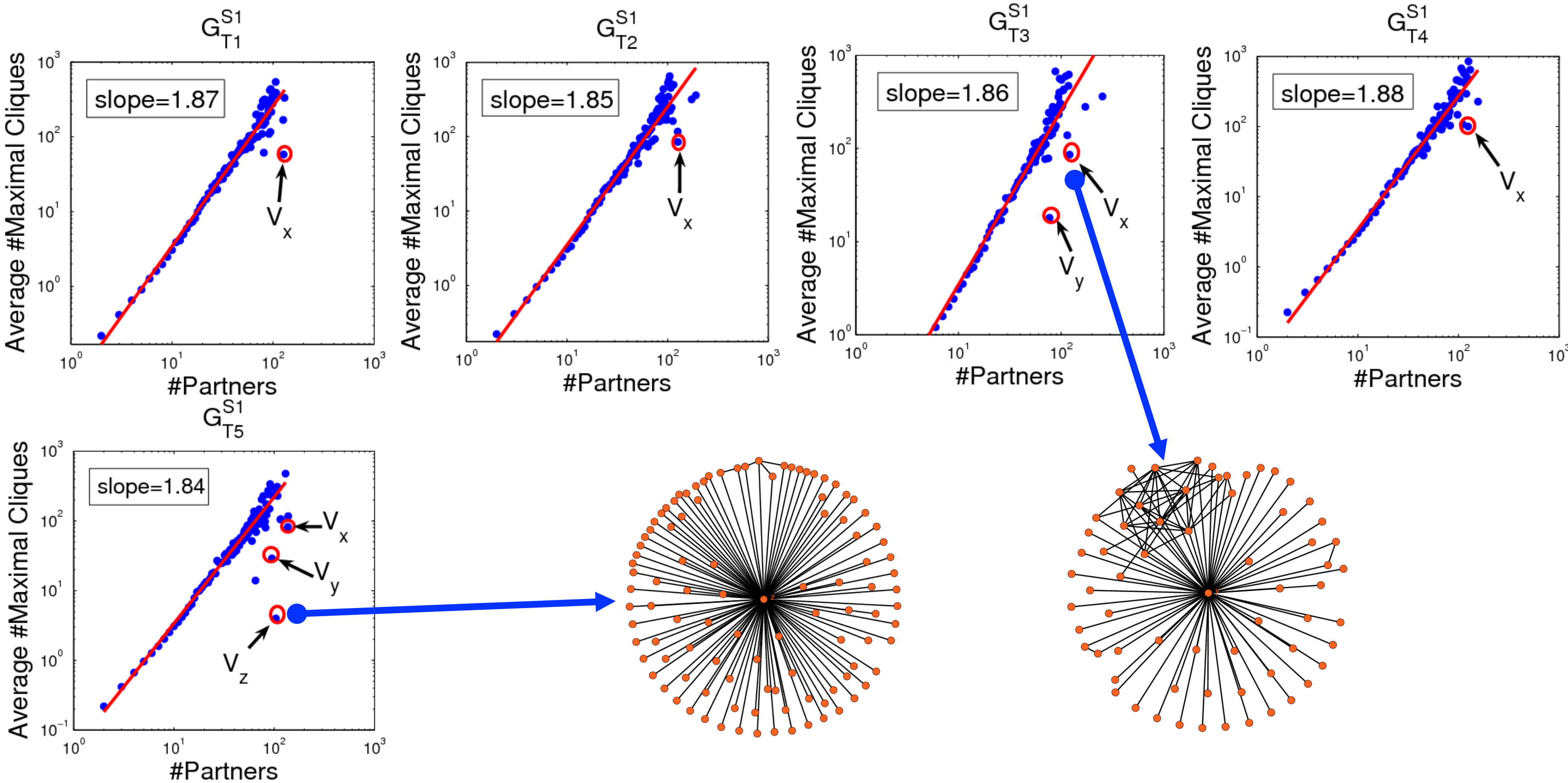
S5: Clique-Degree Power-Law

- Outlier Detection



S5: Clique-Degree Power-Law

- Outlier Detection



Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
    - Degree, eigenvalues
    - Triangles, cliques
    - Other observations
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

Any other ‘laws’?

Yes!

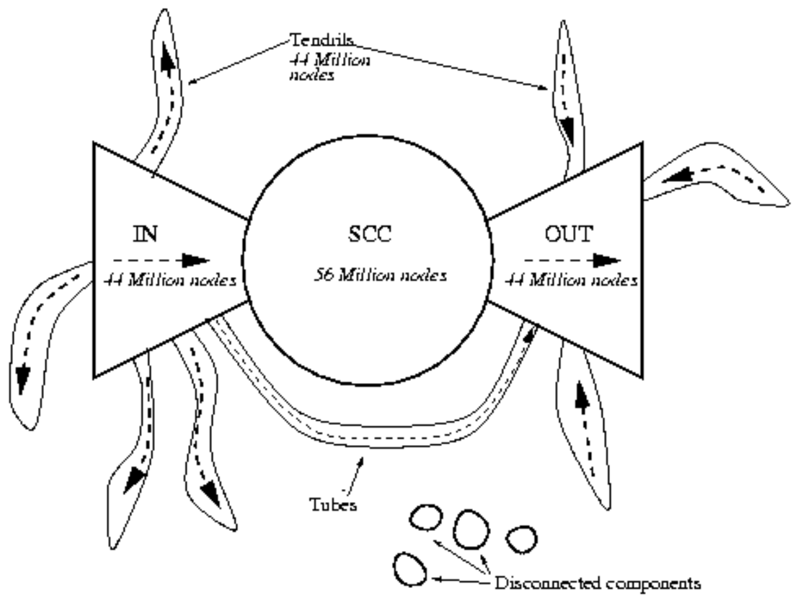
Any other 'laws'?

Yes!

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

Any other 'laws'?

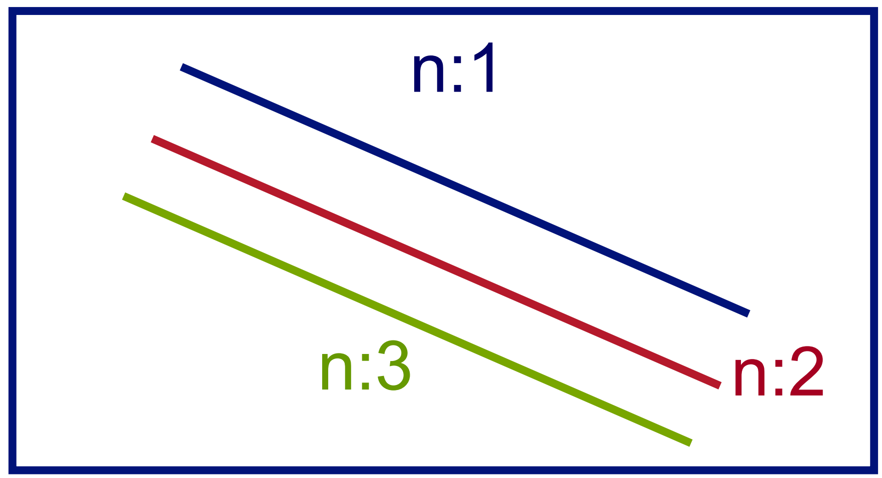
- Bow-tie, for the web [Kumar+ '99]
- IN, SCC, OUT, 'tendrils'
- disconnected components



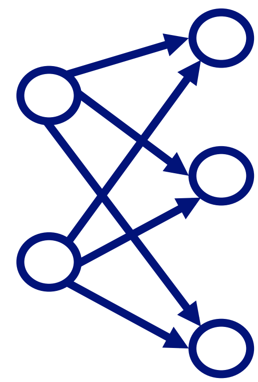
Any other 'laws'?

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

Log(count)



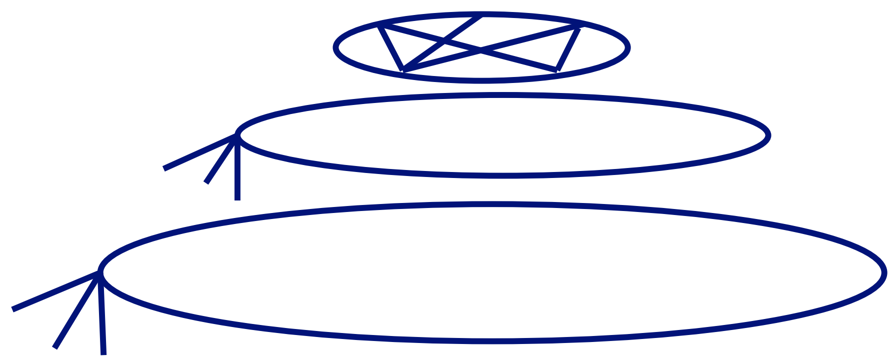
Log(m)



2:3 core
(m:n core)

Any other 'laws'?

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



Carnegie Mellon

Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
    - Degree, eigenvalues
    - Triangles, cliques
    - Other observations
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

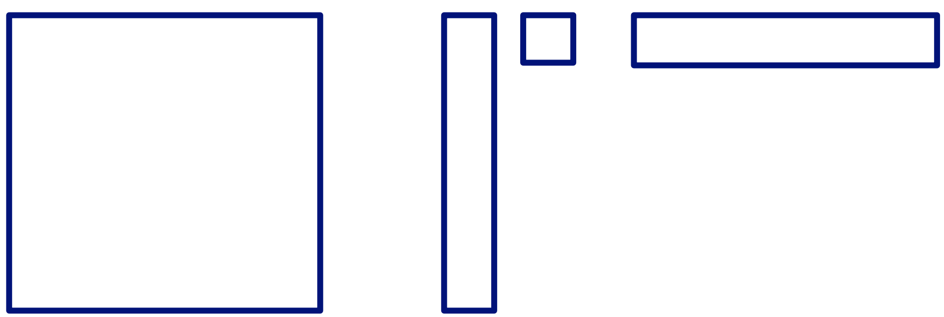
EigenSpokes

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

EigenSpokes

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

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

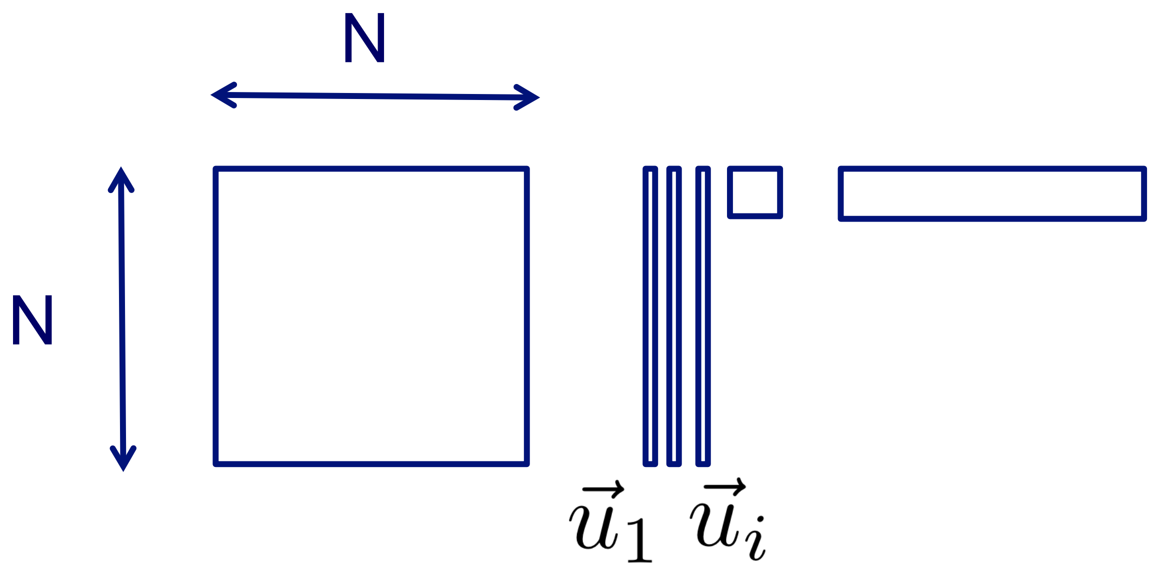


details

EigenSpokes

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

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

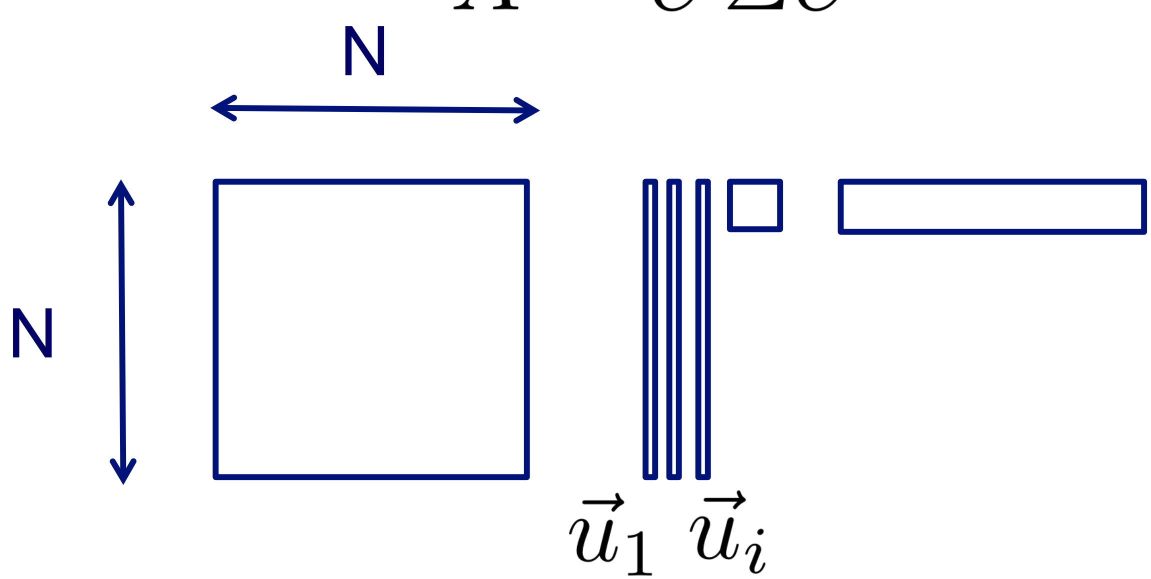


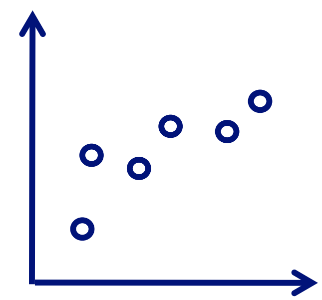
details

EigenSpokes

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

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



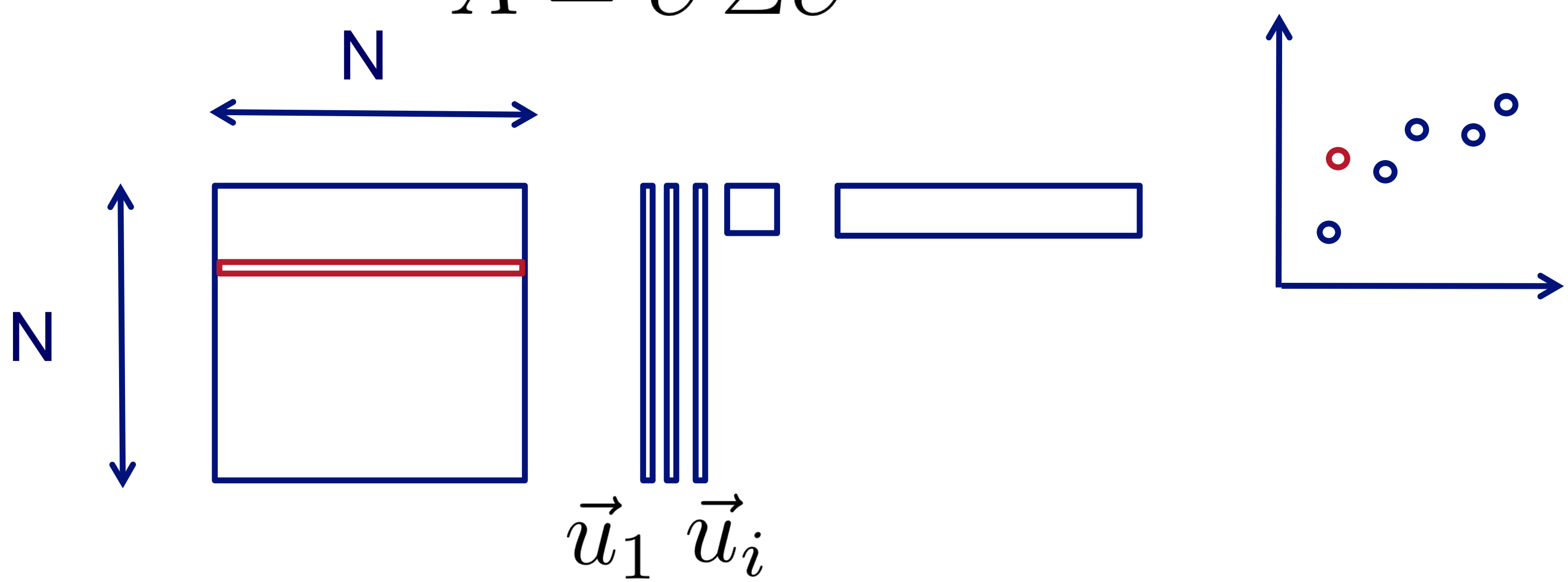


details

EigenSpokes

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

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

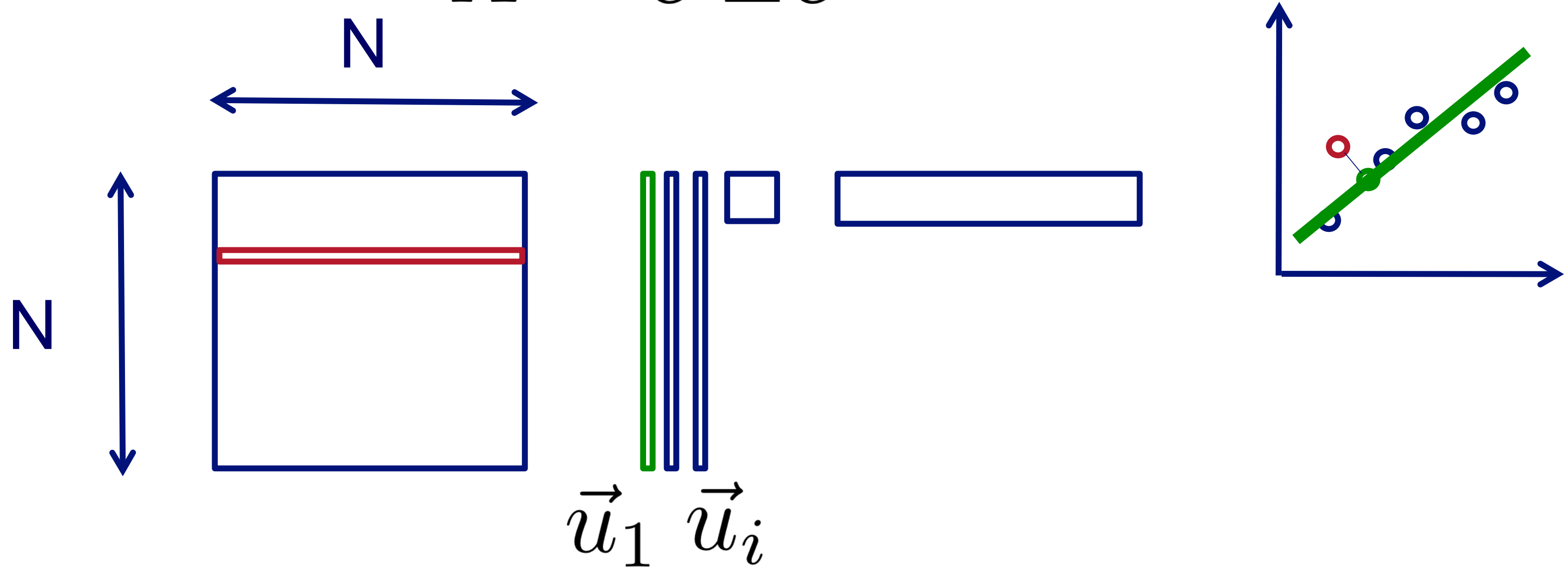


details

EigenSpokes

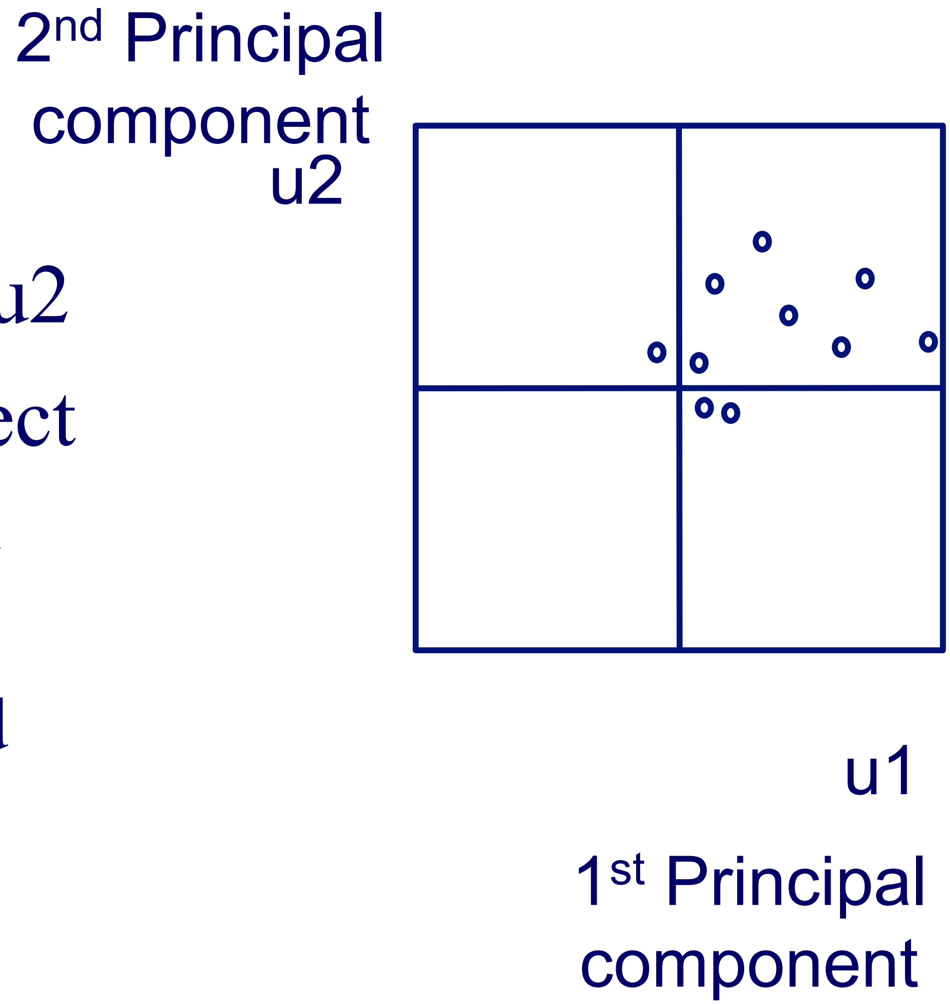
- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

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



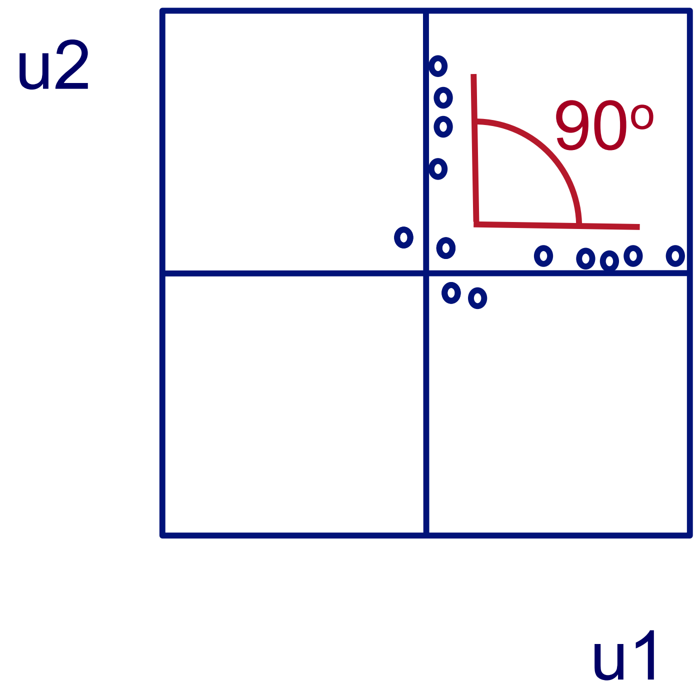
EigenSpokes

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



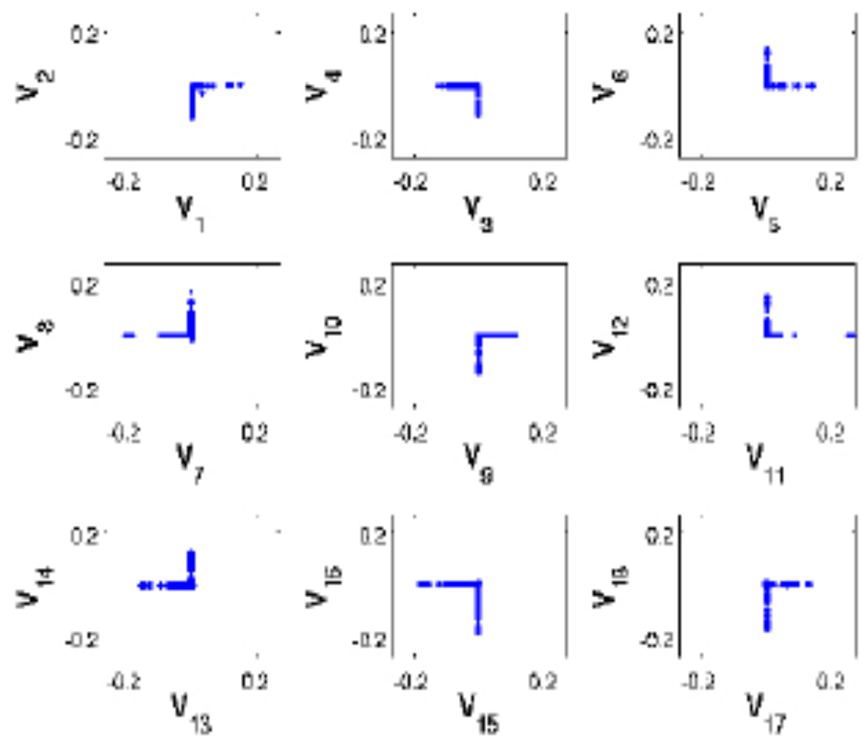
EigenSpokes

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

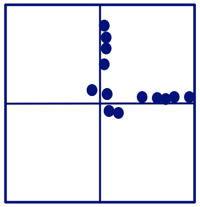


EigenSpokes - pervasiveness

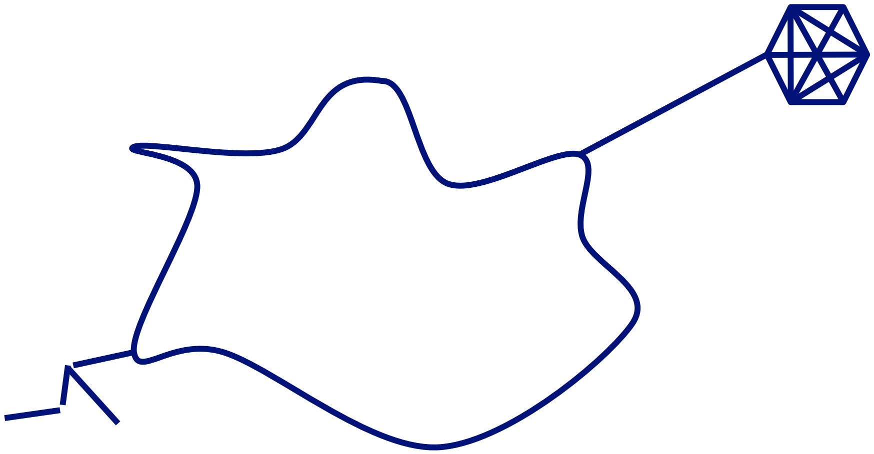
- Present in mobile social graph
  - across time and space
- Patent citation graph



EigenSpokes - explanation

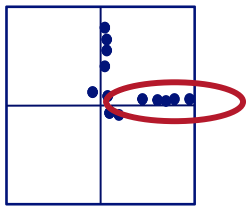


Near-cliques, or near-bipartite-cores, loosely connected

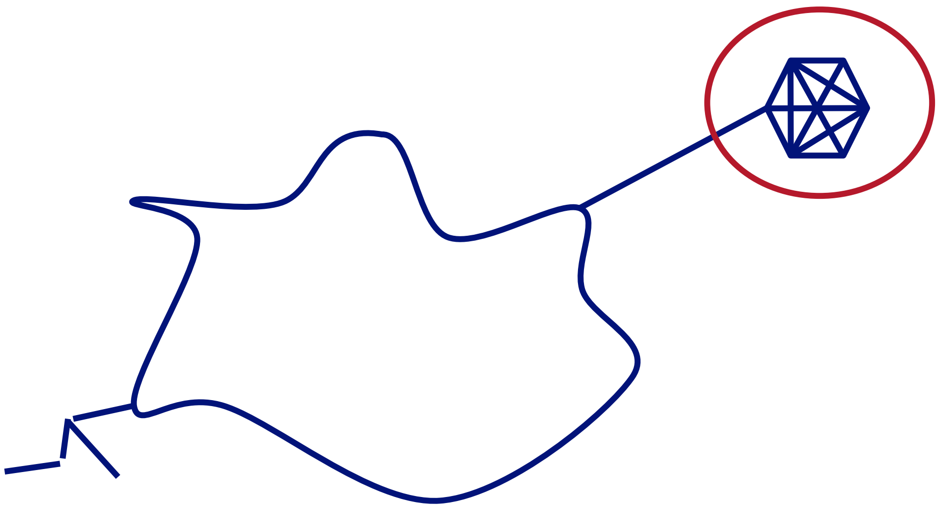


Carnegie Mellon

EigenSpokes - explanation

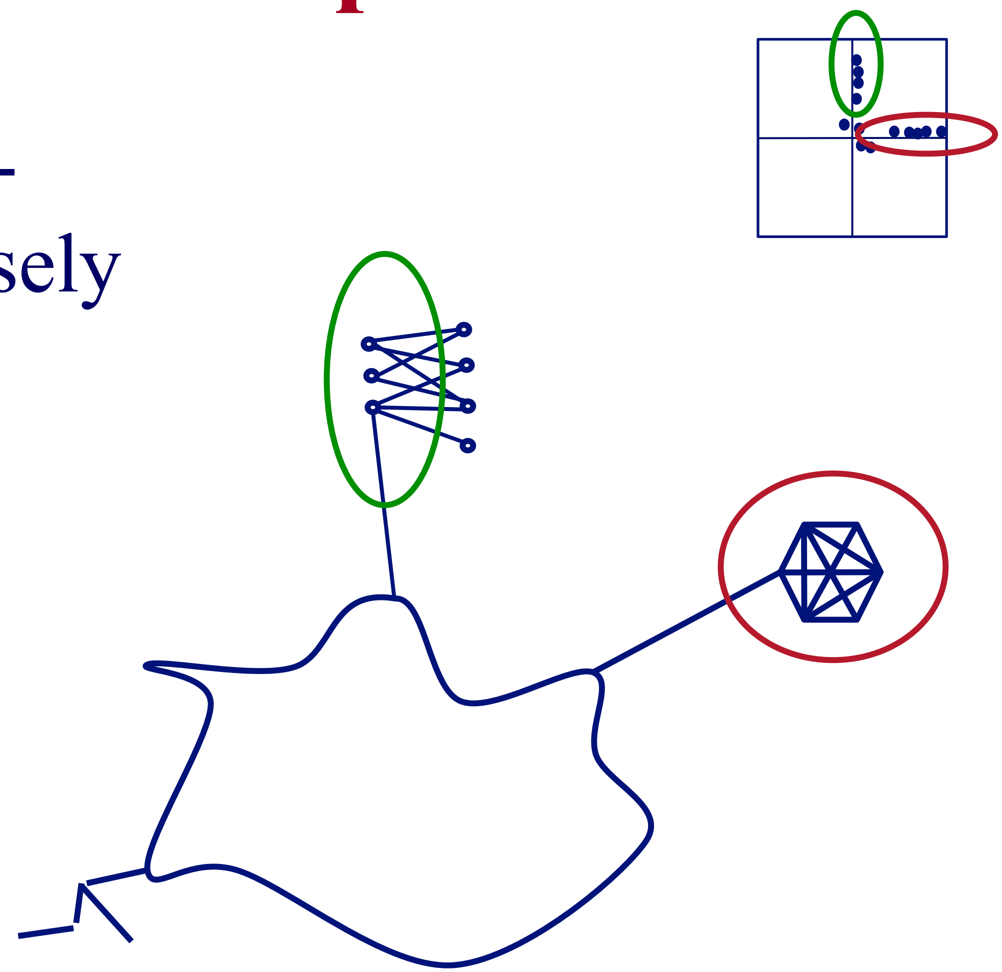


Near-cliques, or near-bipartite-cores, loosely connected



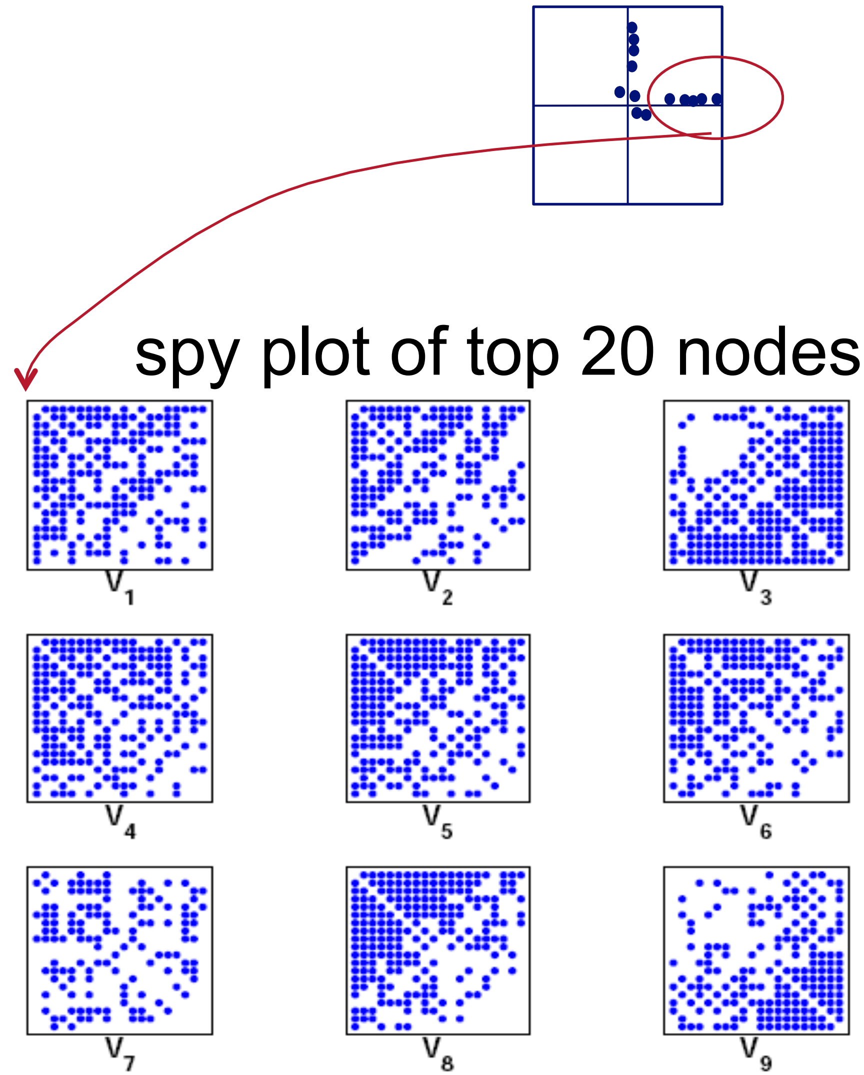
EigenSpokes - explanation

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



EigenSpokes - explanation

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



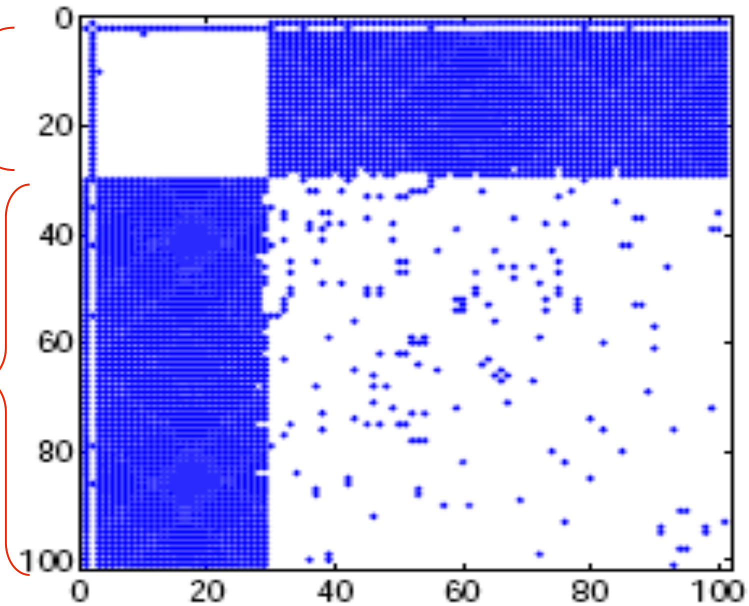
So what?

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

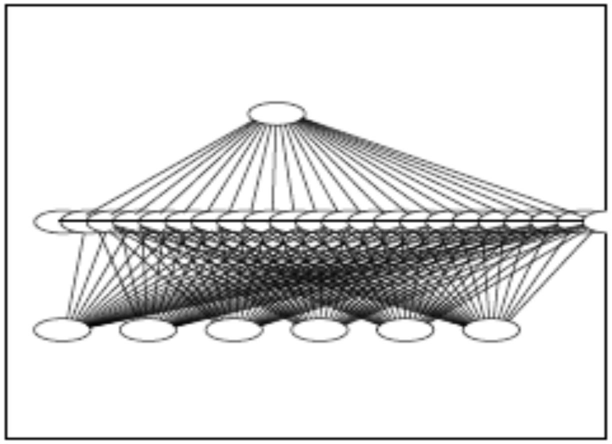
Bipartite Communities!

patents from same inventor(s)

cut-and-paste bibliography!



magnified bipartite community



Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

Observations on weighted graphs?

- A: yes - even more 'laws'!


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

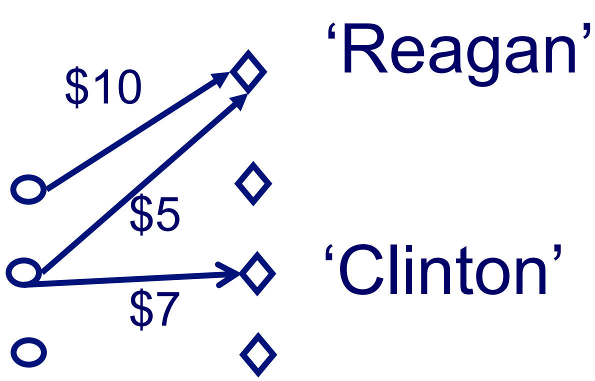
Observation W.1: Fortification

Q: How do the weights

of nodes relate to degree?

Observation W.1: Fortification

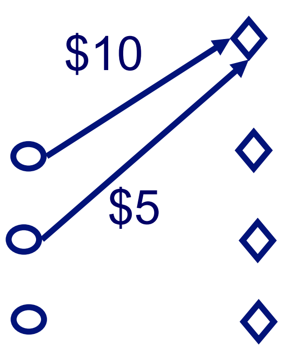
More donors, more $?



Observation W.1: fortification: Snapshot Power Law

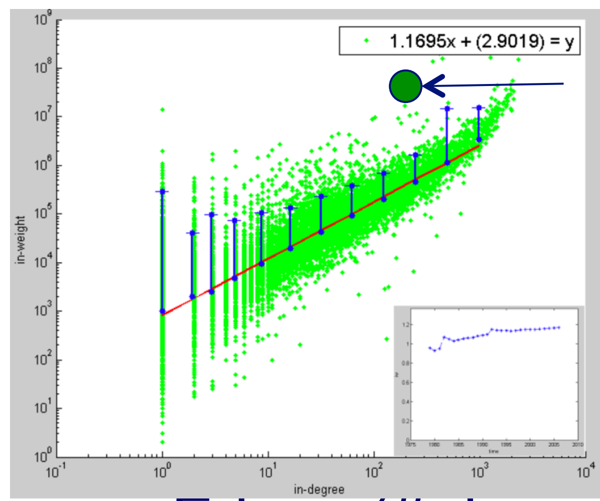
- Weight: super-linear on in-degree
- exponent 'iw': 1.01 < iw < 1.26

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



Orgs-Candidates

In-weights ($)



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

Edges (# donors)

Outline

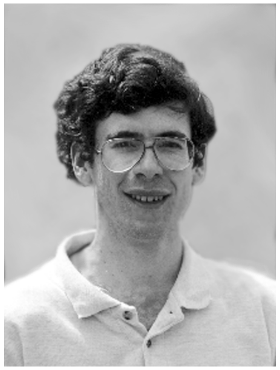
- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

Carnegie Mellon

Problem: Time evolution

- with Jure Leskovec (CMU -> Stanford)

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

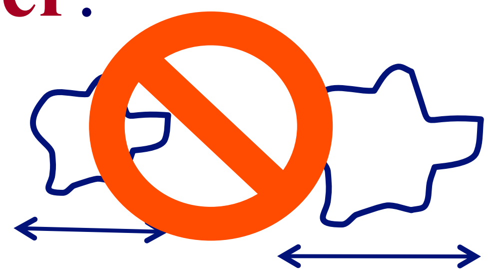


T.1 Evolution of the Diameter

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

T.1 Evolution of the Diameter

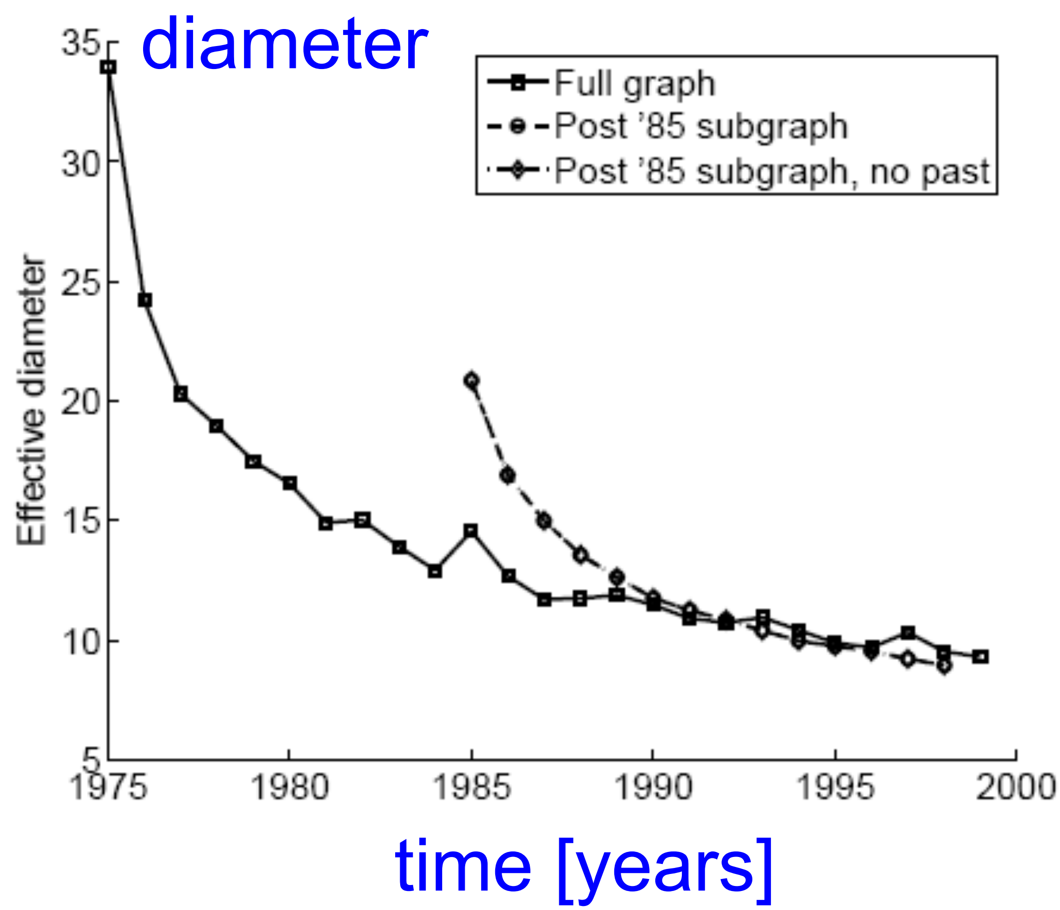
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)



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

T.1 Diameter – "Patents"

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



T.2 Temporal Evolution of the Graphs

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

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

- Q: what is your guess for

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

T.2 Temporal Evolution of the Graphs

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

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

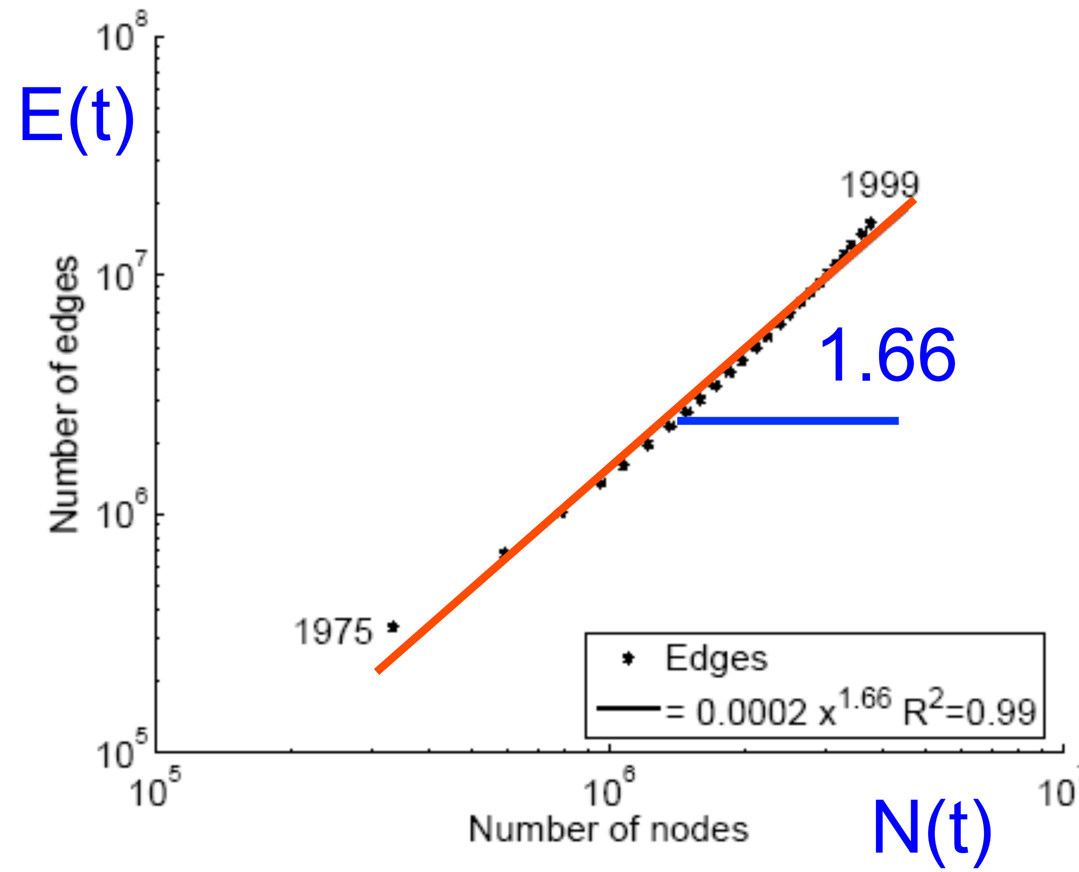
- Q: what is your guess for

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

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

T.2 Densification – Patent Citations

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



Outline

- Introduction – Motivation
- Patterns in graphs
  - Patterns in Static graphs
  - Patterns in Weighted graphs
  - Patterns in Time evolving graphs
- Generators

More on Time-evolving graphs

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

Carnegie Mellon

Observation T.3: NLCC behavior

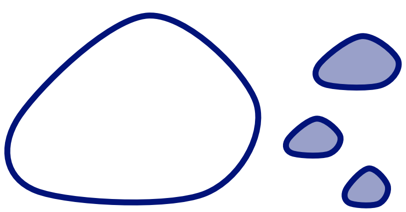
Q: How do NLCC's emerge and join with the GCC?

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

– Do they continue to grow in size?

– or do they shrink?

– or stabilize?



Observation T.3: NLCC behavior

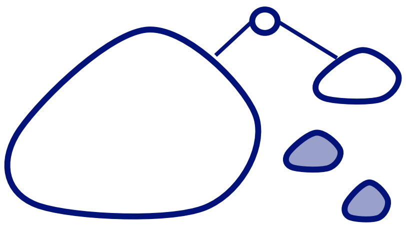
Q: How do NLCC's emerge and join with the GCC?

(``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?

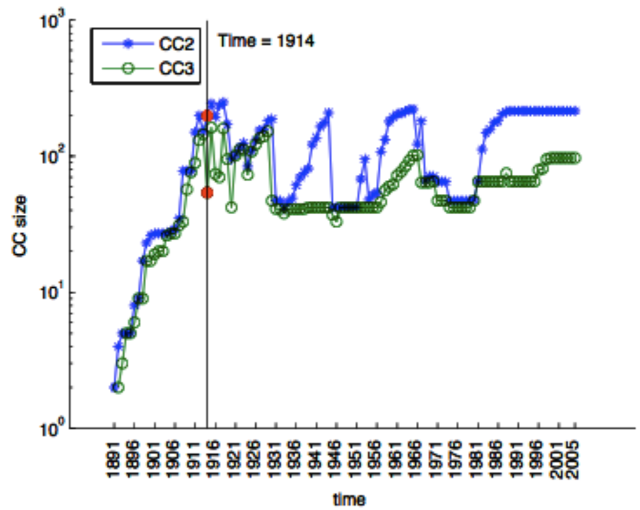
Carnegie Mellon

Observation T.3: NLCC behavior

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

IMDB

CC size



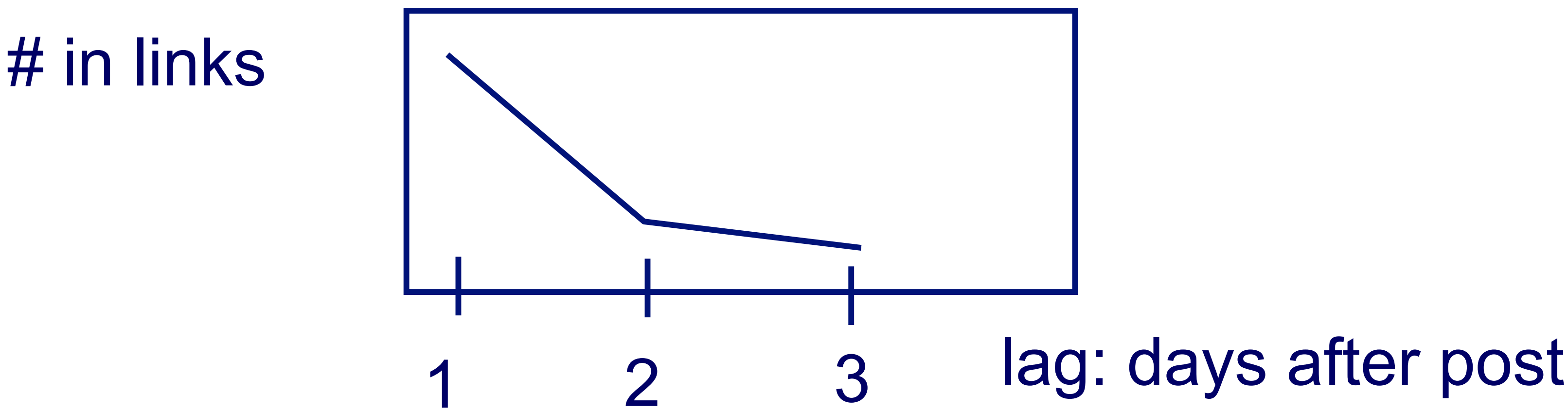
Time-stamp

Timing for Blogs

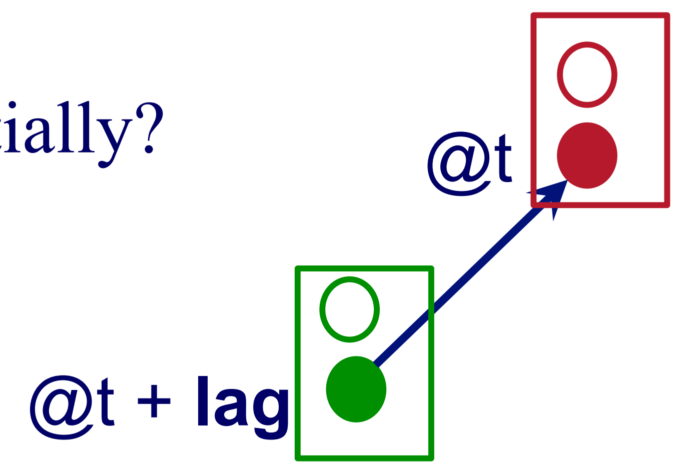
- with Mary McGlohon (CMU)
- Jure Leskovec (CMU->Stanford)
- Natalie Glance (now at Google)
- Mat Hurst (now at MSR)

[SDM'07]

T.4 : popularity over time



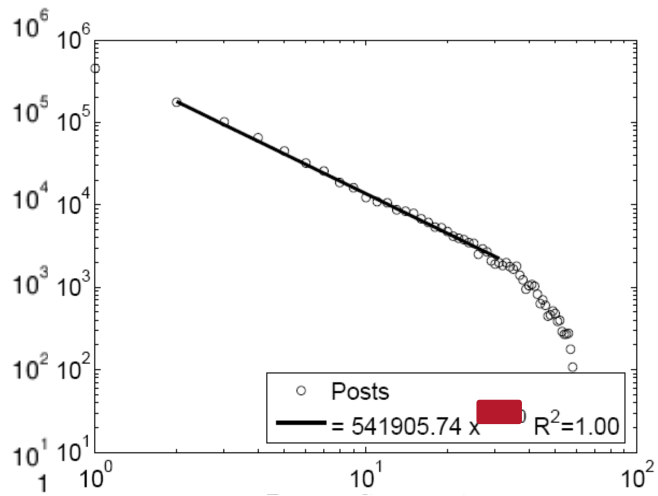
Post popularity drops-off – exponentially?



Carnegie Mellon

T.4 : popularity over time

in links (**log**)



days after post (**log**)

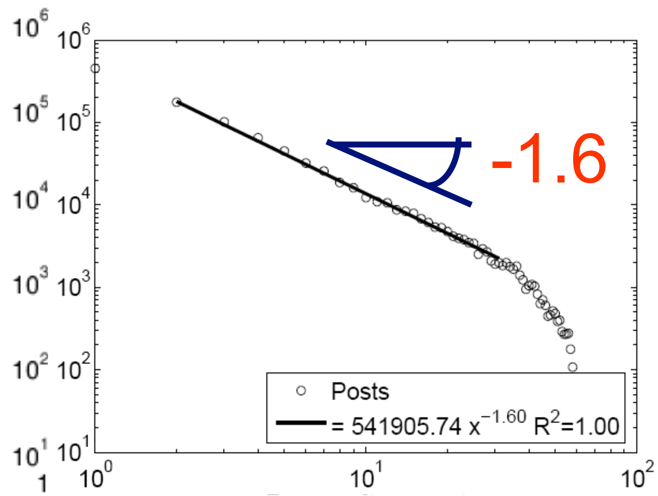
Post popularity drops-off – exponentially?

POWER LAW!

Exponent?

T.4 : popularity over time

in links (**log**)



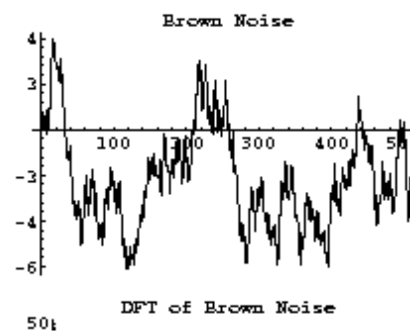
days after post (**log**)

Post popularity drops-off – exponentially?

POWER LAW!

Exponent? -1.6

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



CarnegieMellon

-1.5 slope

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

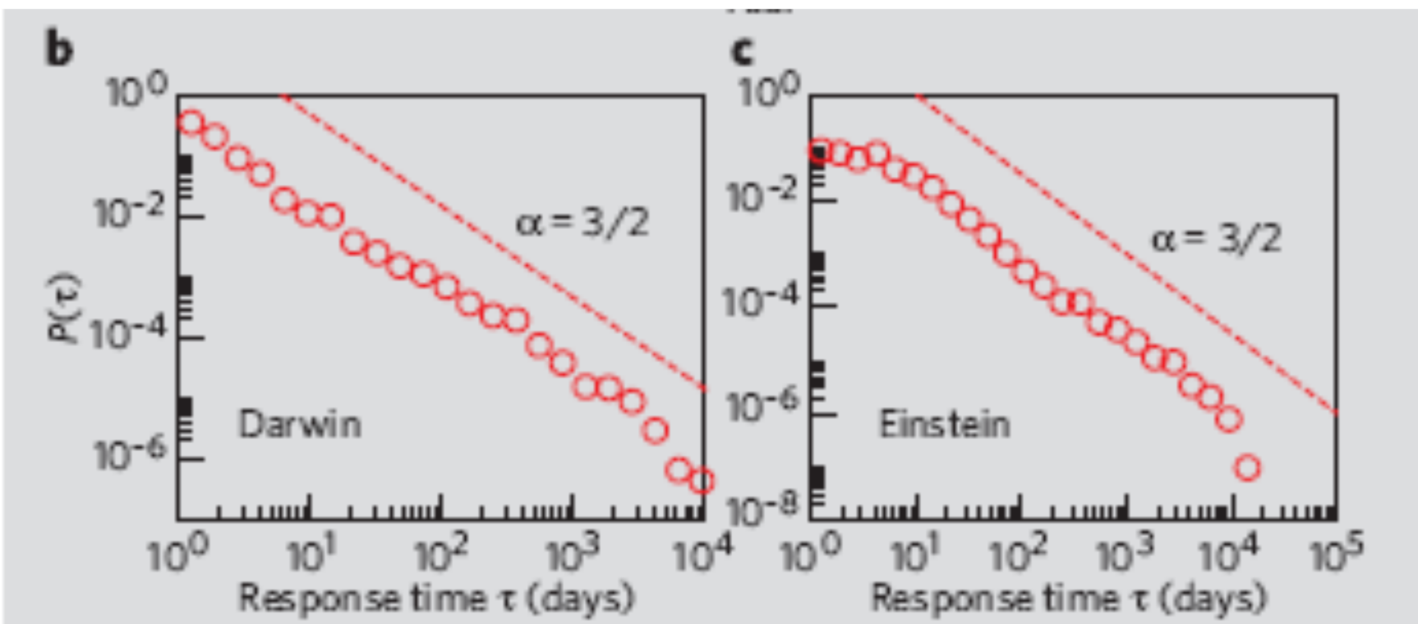


Figure 1 | The correspondence patterns of Darwin and Einstein.

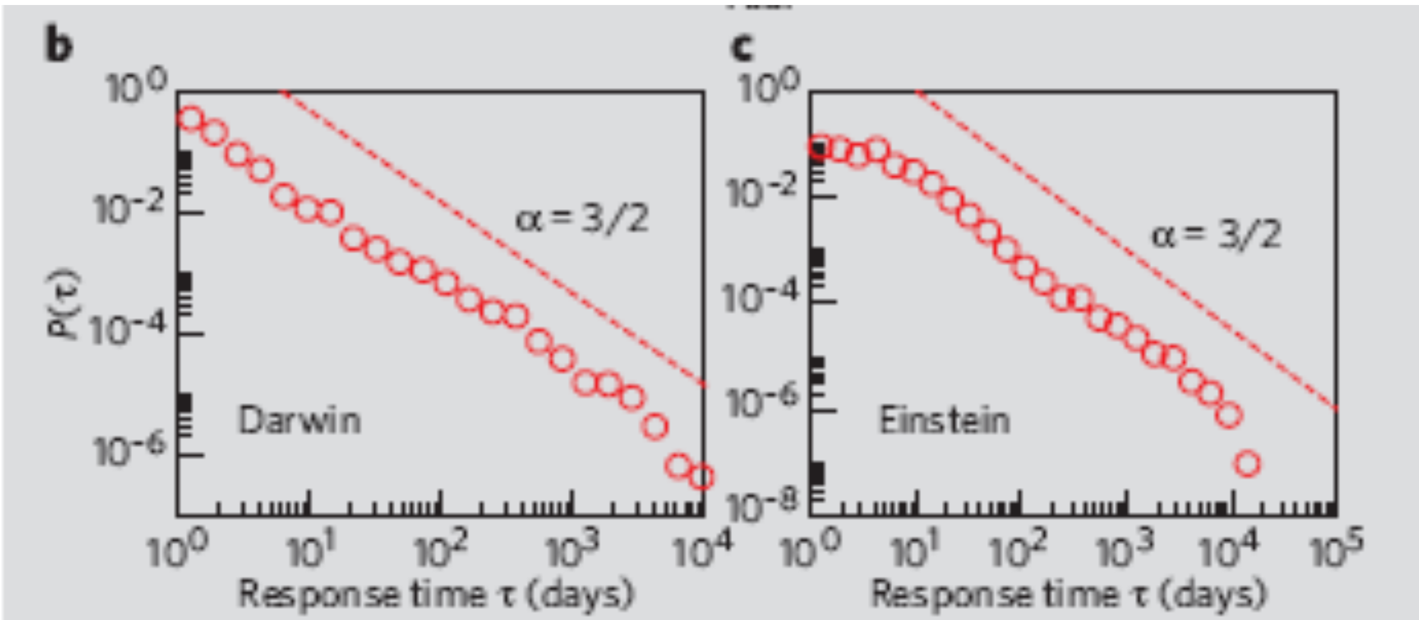


Figure 1 | The correspondence patterns of Darwin and Einstein.

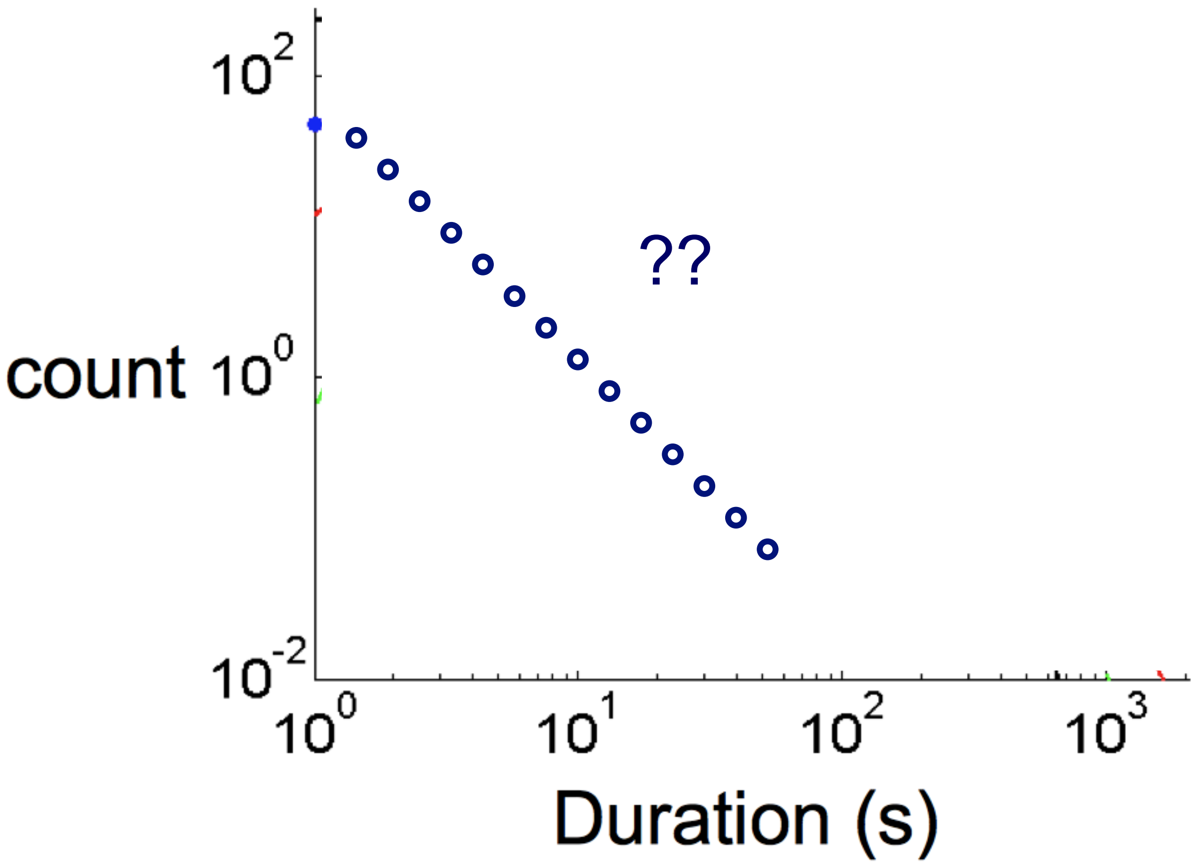
T.5: duration of phonecalls

Surprising Patterns for the Call Duration Distribution of Mobile Phone Users

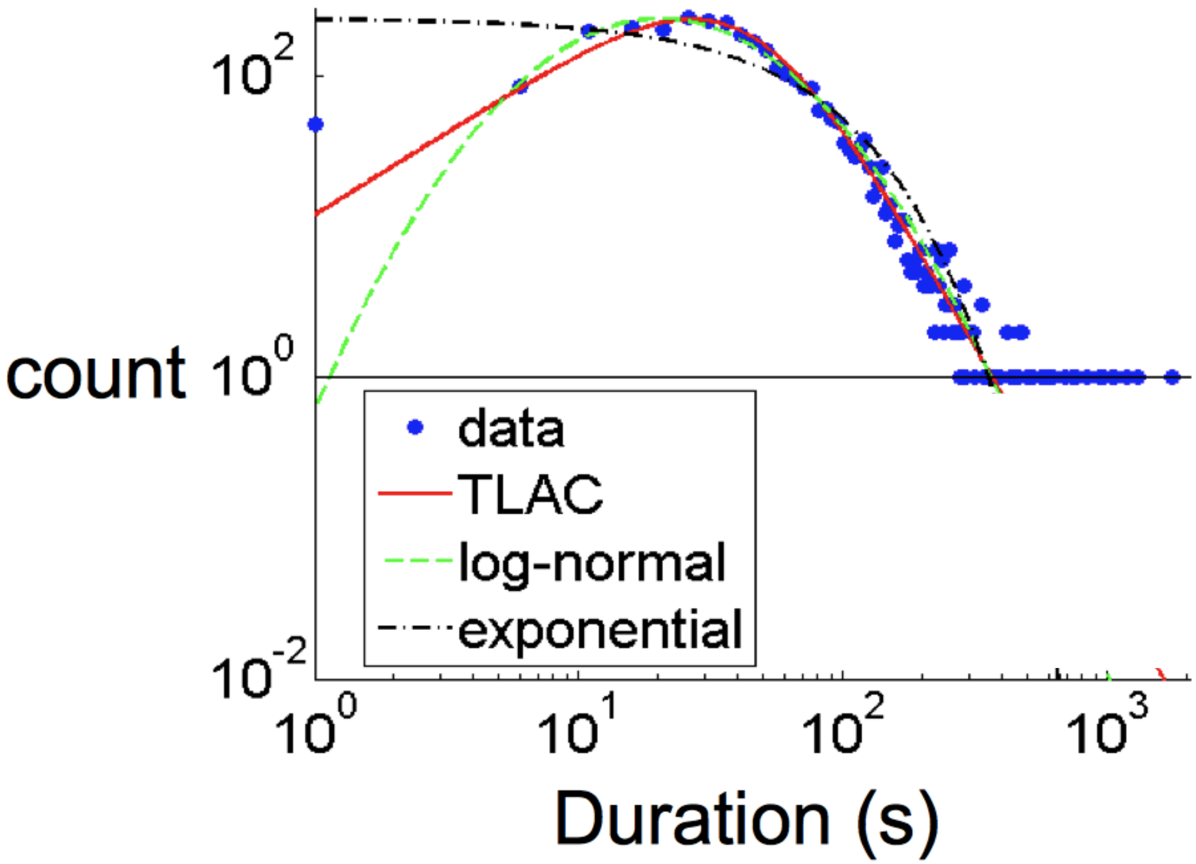
Pedro O. S. Vaz de Melo, Leman Akoglu, Christos Faloutsos, Antonio A. F. Loureiro

PKDD 2010

Probably, power law (?)

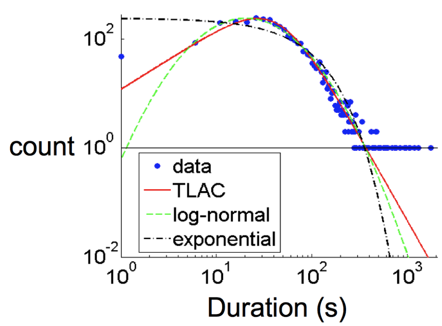


No Power Law!


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

'TLaC: Lazy Contractor'

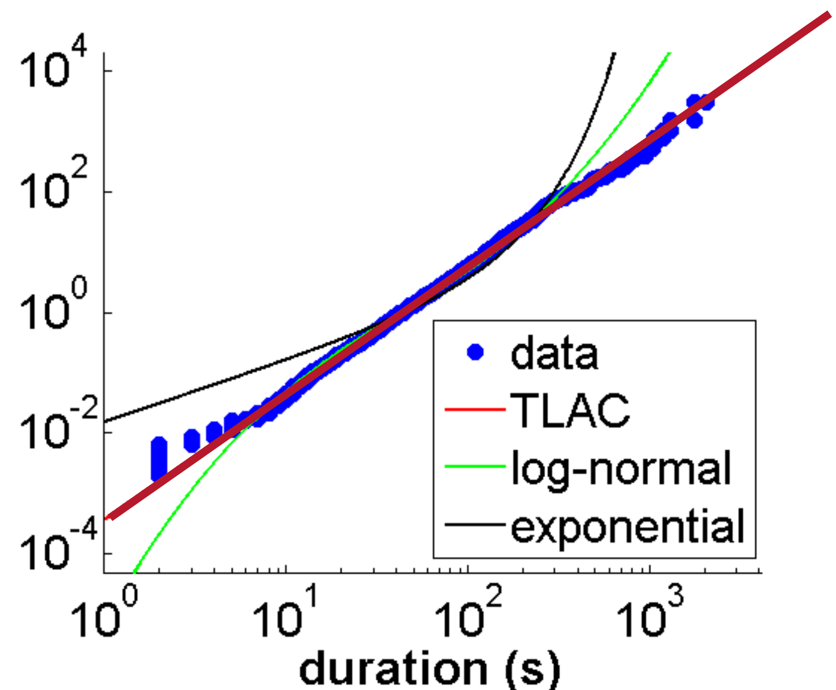
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

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

== power law



Carnegie Mellon

Data Description

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

Outline

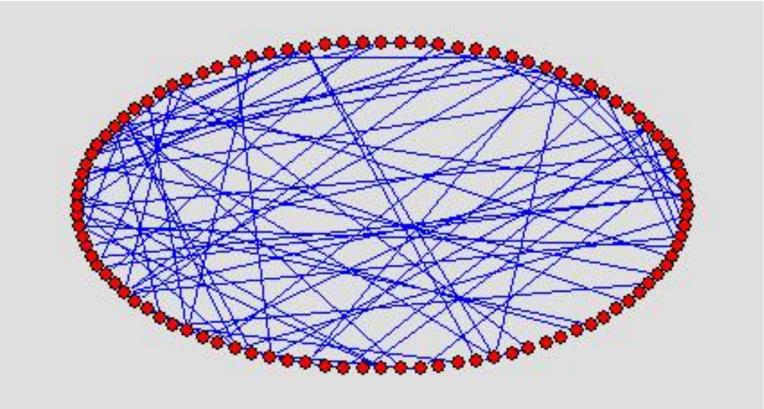
- Introduction – Motivation
- Patterns in graphs
- Generators
  - Erdos-Renyi
  - Degree based
  - Process based
  - Kronecker

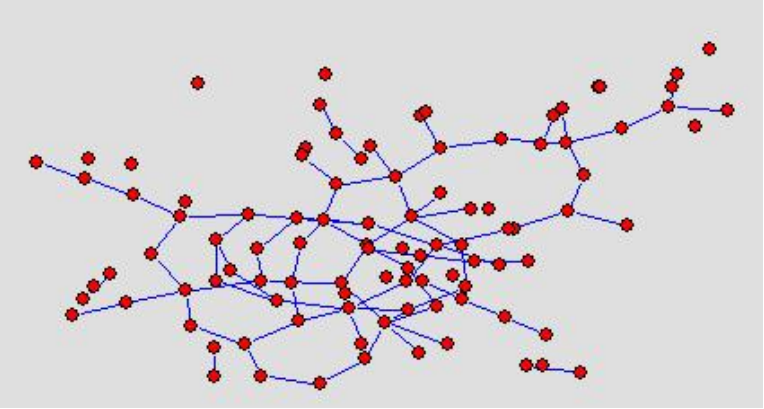
Generators

- How to generate random, realistic graphs?
  - Erdos-Renyi model: beautiful, but unrealistic
  - degree-based generators
  - process-based generators
  - recursive/self-similar generators

Erdos-Renyi

- random graph – 100 nodes, avg degree = 2
- Fascinating properties (phase transition)
- But: unrealistic (Poisson degree distribution != power law)



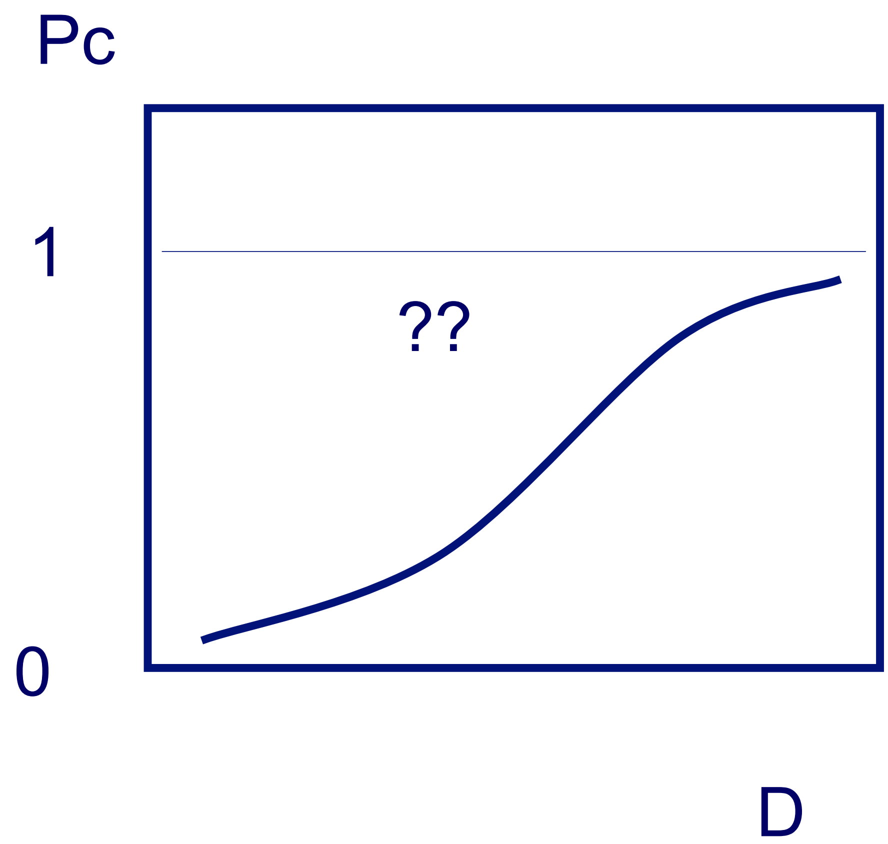


details

E-R model & Phase transition

- vary avg degree D
- watch Pc =

  Prob(there is a giant connected component)
- How do you expect it to be?

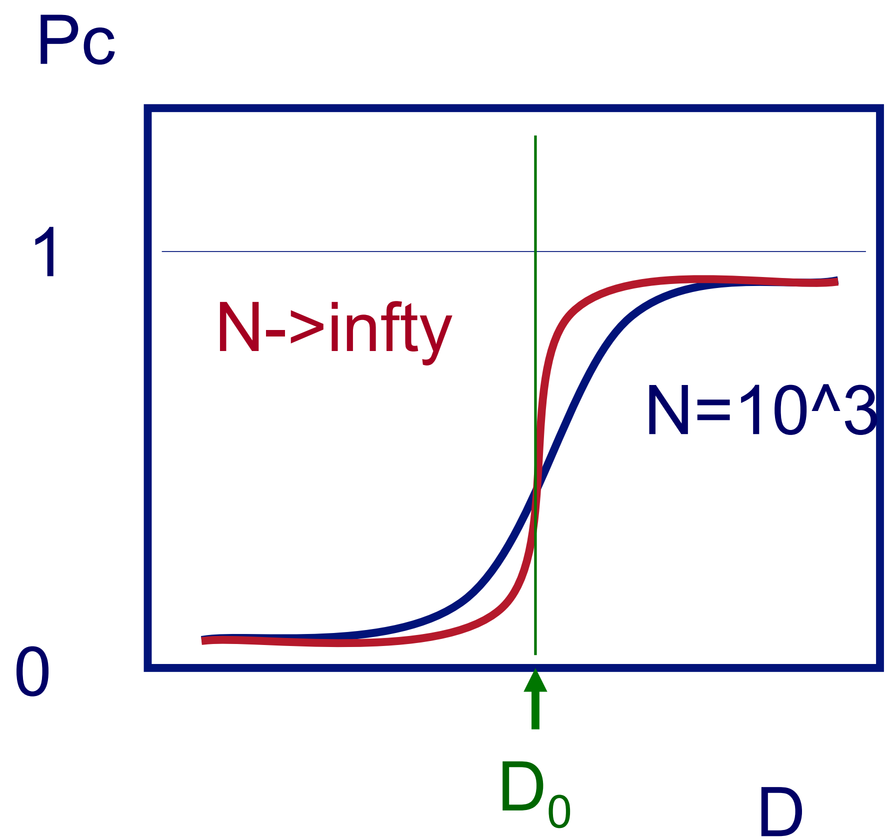


details

E-R model & Phase transition

- vary avg degree D
- watch Pc =

  Prob(there is a giant connected component)
- How do you expect it to be?



Degree-based

- Figure out the degree distribution (eg., 'Zipf')
- Assign degrees to nodes
- Put edges, so that they match the original degree distribution

Process-based

- Barabasi; Barabasi-Albert: Preferential attachment -> power-law tails!
  - ‘rich get richer’
- [Kumar+]: preferential attachment + mimick
  - Create ‘communities’

Process-based (cont'd)

- [Fabrikant+, '02]: H.O.T.: connect to closest, high connectivity neighbor
- [Pennock+, '02]: Winner does NOT take all

Outline

- Introduction – Motivation
- Patterns in graphs
- Generators
  - Erdos-Renyi
  - Degree based
  - Process based
  - Kronecker

Recursive generators

- (RMAT [Chakrabarti+,'04])
- Kronecker product

Wish list for a generator:

- Power-law-tail in- and out-degrees
- Power-law-tail scree plots
- **shrinking/constant** diameter
- Densification Power Law
- communities-within-communities

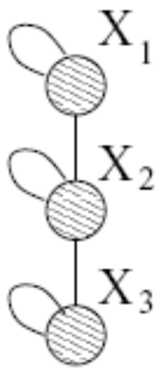
Wish list for a generator:

- Power-law-tail in- and out-degrees
- Power-law-tail scree plots
- **shrinking/constant** diameter
- Densification Power Law
- communities-within-communities

Q: how to achieve all of them?

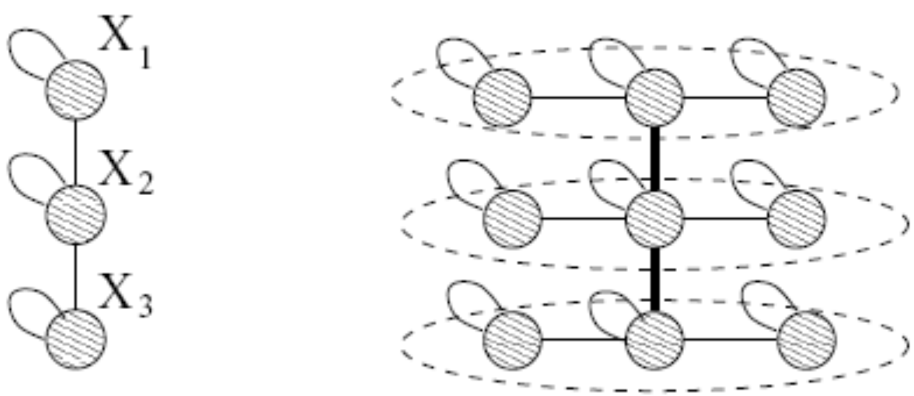
A: Self-similarity - Kronecker matrix product [Leskovec+05b]

Kronecker product



(a) Graph G_1

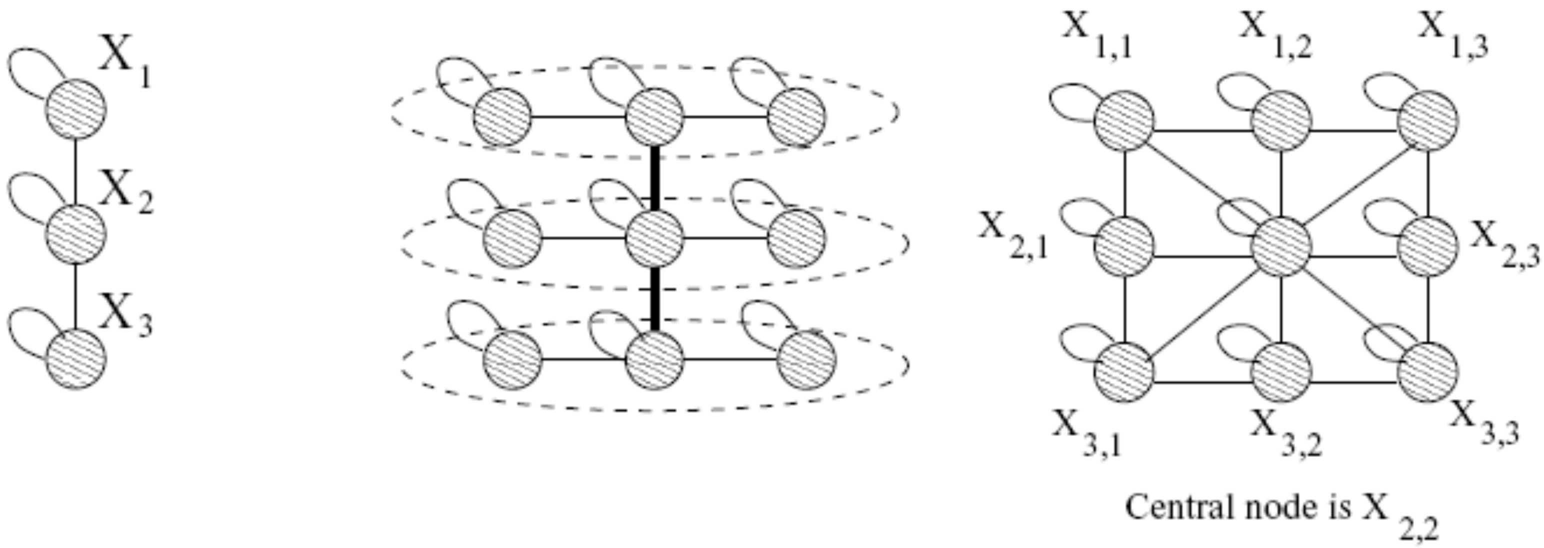
Kronecker product



(a) Graph G_1 (b) Intermediate stage

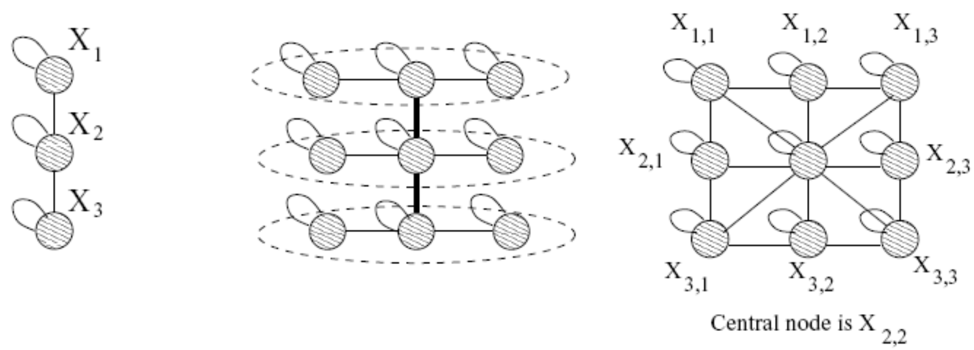
Carnegie Mellon

Kronecker product



(a) Graph G_1 (b) Intermediate stage (c) Graph $G_2 = G_1 \otimes G_1$

Kronecker product



(a) Graph G_1 (b) Intermediate stage (c) Graph $G_2 = G_1 \otimes G_1$

1	1	0
1	1	1
0	1	1

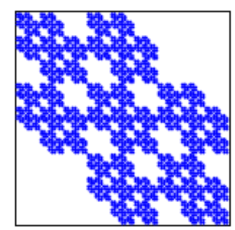
(d) Adjacency matrix of G_1

N

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

(e) Adjacency matrix of $G_2 = G_1 \otimes G_1$

N*N



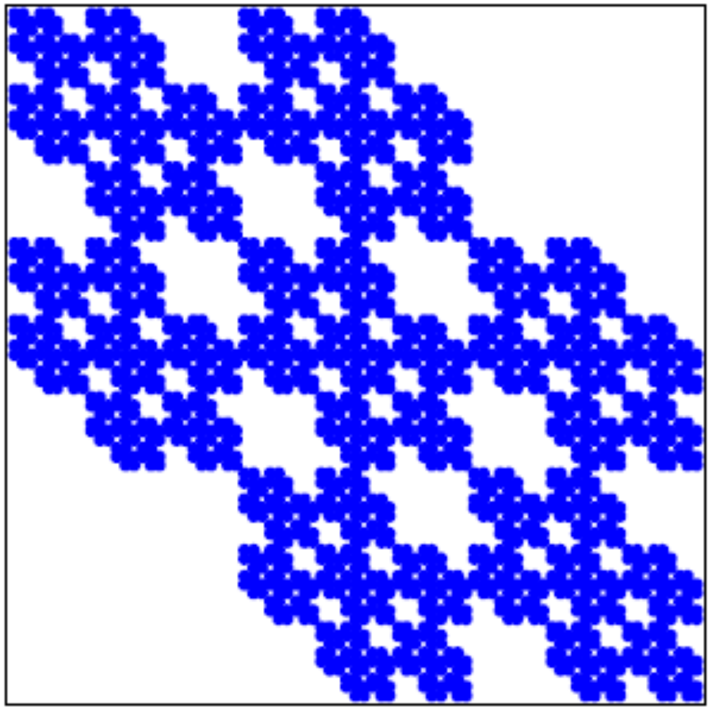
(f) Plot of G_4

N**4

Carnegie Mellon

Kronecker Product – a Graph

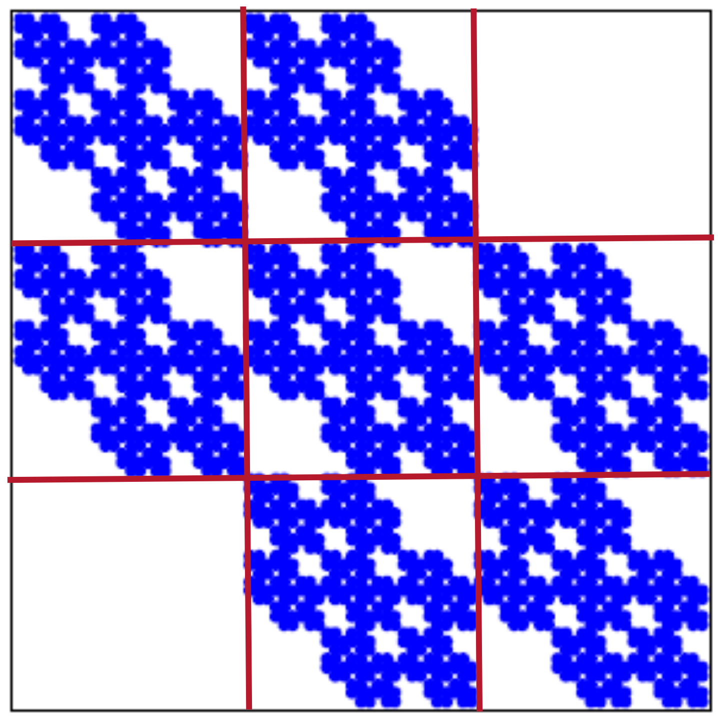
- Continuing multiplying with G_1 we obtain G_4 and so on …



G_4 adjacency matrix

Kronecker Product – a Graph

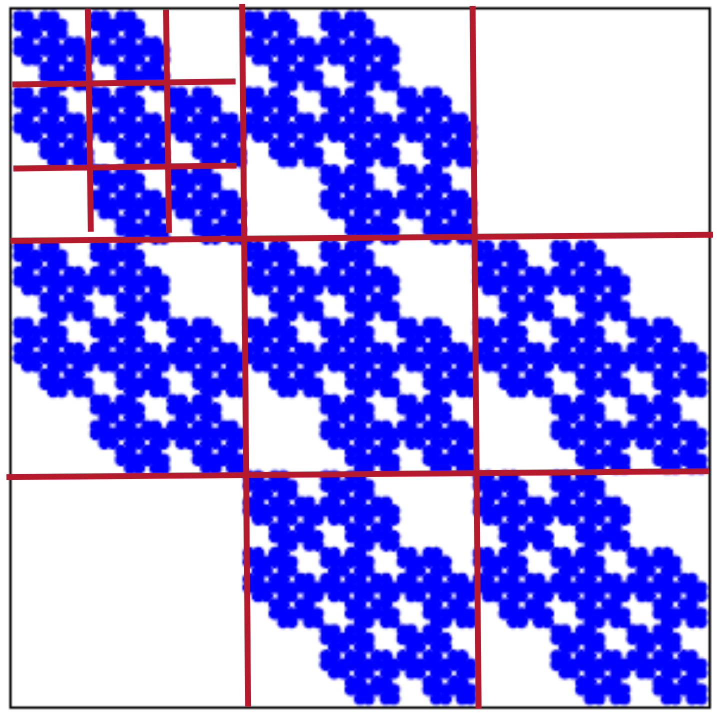
- Continuing multiplying with G_1 we obtain G_4 and so on …



G_4 adjacency matrix

Kronecker Product – a Graph

- Continuing multiplying with G_1 we obtain G_4 and so on …



G_4 adjacency matrix

Carnegie Mellon

Properties of Kronecker graphs:

- ✓ Power-law-tail in- and out-degrees
- ✓ Power-law-tail scree plots
- ✓ **constant** diameter
- ✓ perfect Densification Power Law
- ✓ communities-within-communities

Properties of Kronecker graphs:

- ✓ Power-law-tail in- and out-degrees
- ✓ Power-law-tail scree plots
- ✓ **constant** diameter
- ✓ perfect Densification Power Law
- ✓ communities-within-communities

and we can <u>prove all</u> of the above

(**first** generator that does that)

Carnegie Mellon

skip

Stochastic Kronecker Graphs

- Create $N_1 \times N_1$ probability matrix P_1
- Compute the k^{th} Kronecker power P_k
- For each entry p_{uv} of P_k include an edge (u,v) with probability p_{uv}

0.4	0.2
0.1	0.3

P_1

Kronecker multiplication →

0.16	0.08	0.08	0.04
0.04	0.12	0.02	0.06
0.04	0.02	0.12	0.06
0.01	0.03	0.03	0.09

P_k

→ Instance Matrix G_2

flip biased coins

Experiments

- How well can we match real graphs?
  - Arxiv: physics citations:
    - 30,000 papers, 350,000 citations
    - 10 years of data
  - U.S. Patent citation network
    - 4 million patents, 16 million citations
    - 37 years of data
  - Autonomous systems – graph of internet
    - Single snapshot from January 2002
    - 6,400 nodes, 26,000 edges
- We show both static and temporal patterns

(Q: how to fit the parm's?)

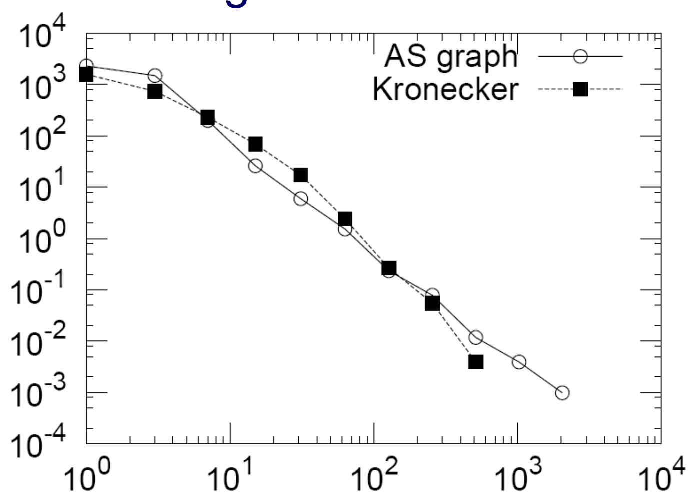
A:

- Stochastic version of Kronecker graphs +
- Max likelihood +
- Metropolis sampling
- [Leskovec+, ICML'07]

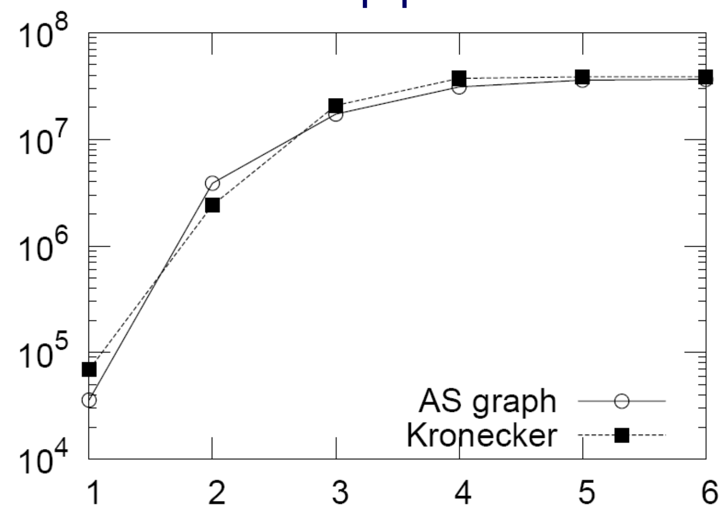
Carnegie Mellon

Experiments on real AS graph

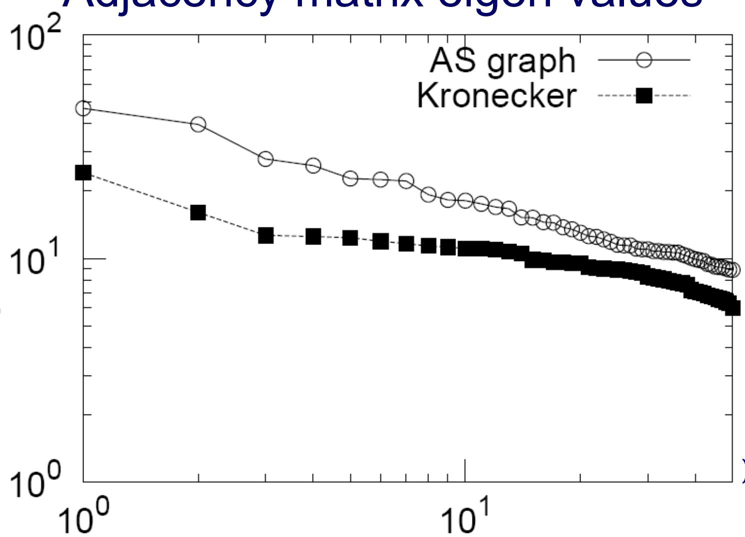
Degree distribution



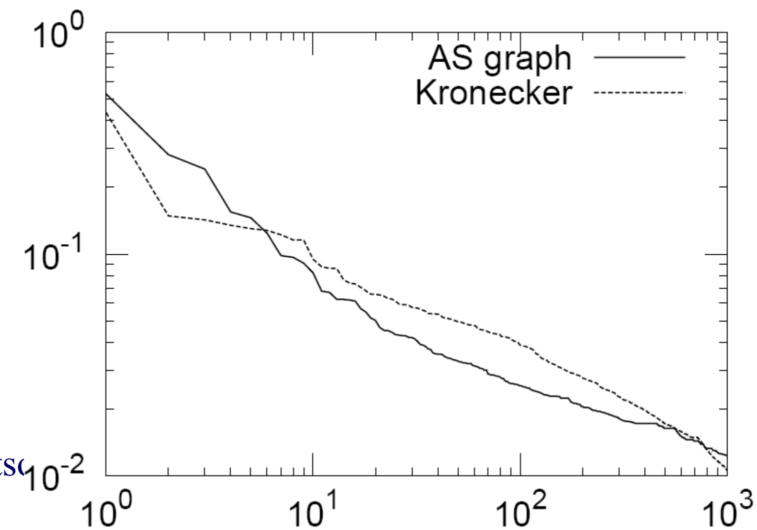
Hop plot



Adjacency matrix eigen values



Network value



Conclusions

- Kronecker graphs have:
  - All the static properties
    - ✓ Heavy tailed degree distributions
    - ✓ Small diameter
    - ✓ Multinomial eigenvalues and eigenvectors
  - All the temporal properties
    - ✓ Densification Power Law
    - ✓ Shrinking/Stabilizing Diameters
  - We can formally prove these results

Carnegie Mellon

OVERALL CONCLUSIONS

- Several new **patterns** (fortification, triangle-laws, conn. components, etc)
- Recursive generators (Kronecker), with provable properties

Carnegie Mellon

References

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

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

References

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

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

References

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

References

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

Carnegie Mellon

References

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

References

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

References

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

References

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

References

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

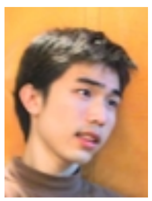
Project info

www.cs.cmu.edu/~pegasus

Chau, Polo

McGlohon, Mary

Tsourakakis, Babis




Akoglu, Leman

Kang, U

Prakash, Aditya

Tong, Hanghang

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

END

Extra material – why so many power laws?

At least 6-7 mechanisms (!)

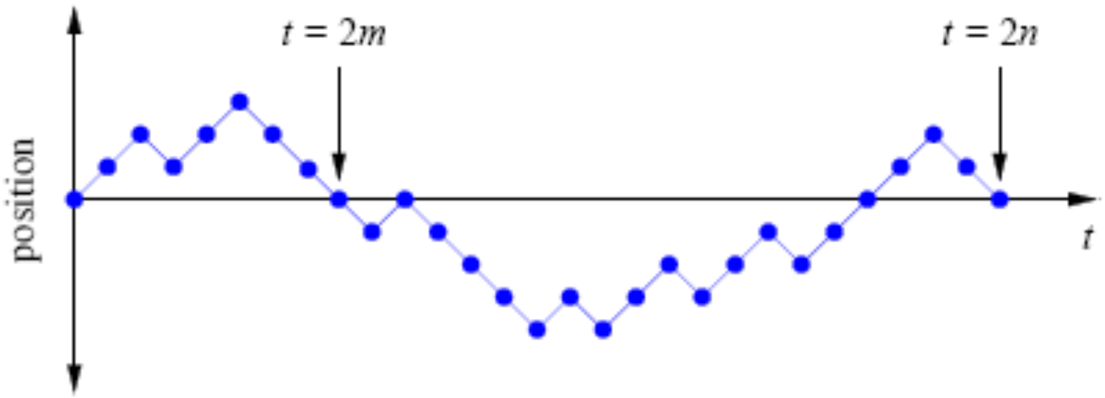
- Power laws, Pareto distributions and Zipf's law Contemporary Physics 46, 323-351 (2005)

Outline

- Generative mechanisms
  - Random walk
  - Yule distribution = CRP
  - Percolation
  - Self-organized criticality
  - Other

Carnegie Mellon

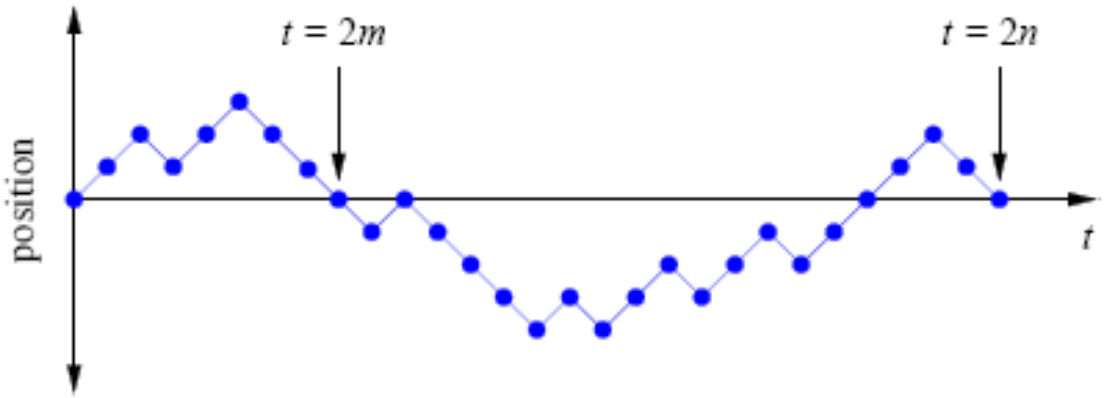
Random walks



Inter-arrival times PDF: *p(t)* ~ ?? ^{2}

Carnegie Mellon

Random walks



Inter-arrival times PDF: $p(t) \sim t^{-3/2}$

Random walks

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

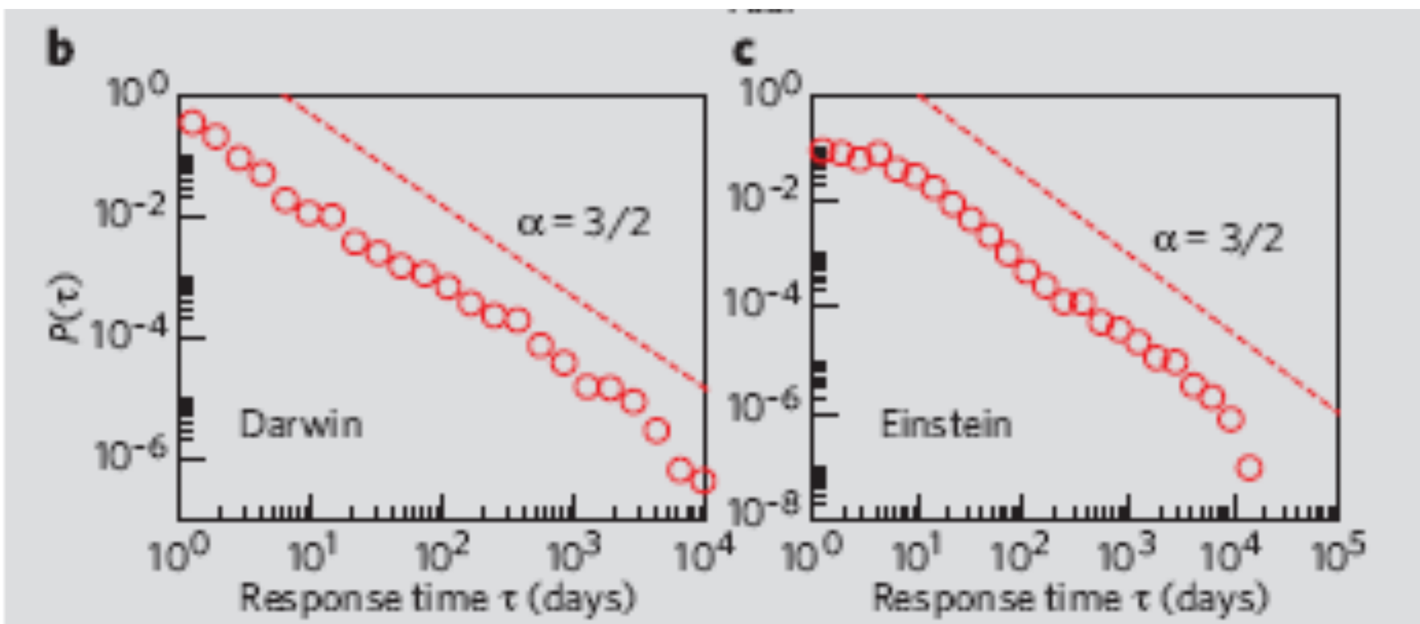


Figure 1 | The correspondence patterns of Darwin and Einstein.

Outline

- Generative mechanisms
  - Random walk
  - Yule distribution = CRP
  - Percolation
  - Self-organized criticality
  - Other

Yule distribution and CRP

Chinese Restaurant Process (CRP):

Newcomer to a restaurant

- Joins an existing table (preferring large groups
- Or starts a new table/group of its own, with prob *1/m*

a.k.a.: rich get richer; Yule process

Yule distribution and CRP

Then:

Prob(k people in a group) = p_k

$= (1 + 1/m)\, B(k, 2+1/m)$

$\sim k^{-(2+1/m)}$

(since B(a,b) ~ a ** (-b) : power law tail)

Yule distribution and CRP

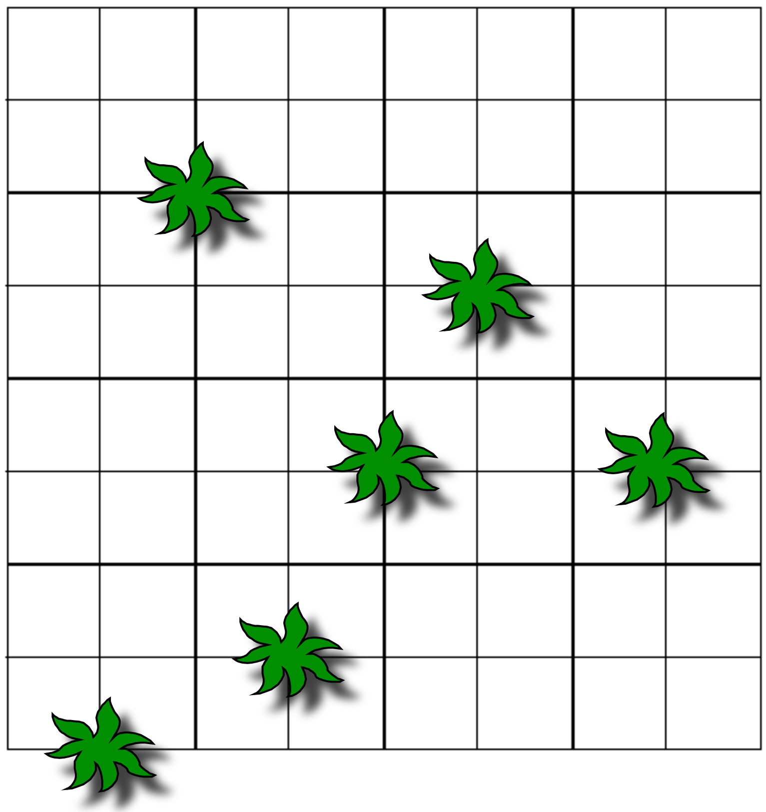
- Yule process
- Gibrat principle
- Matthew effect
- Cumulative advantage
- Preferential attachement
- ‘rich get richer’

Carnegie Mellon

Outline

- Generative mechanisms
  - Random walk
  - Yule distribution = CRP
  - Percolation
  - Self-organized criticality
  - Other

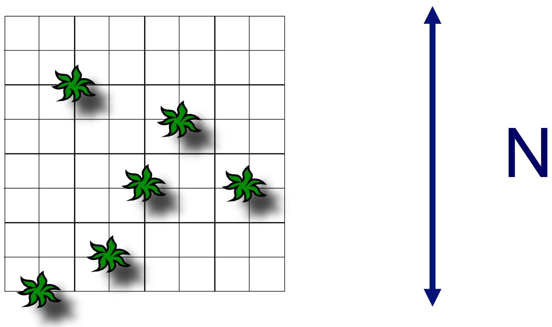
Percolation and forest fires

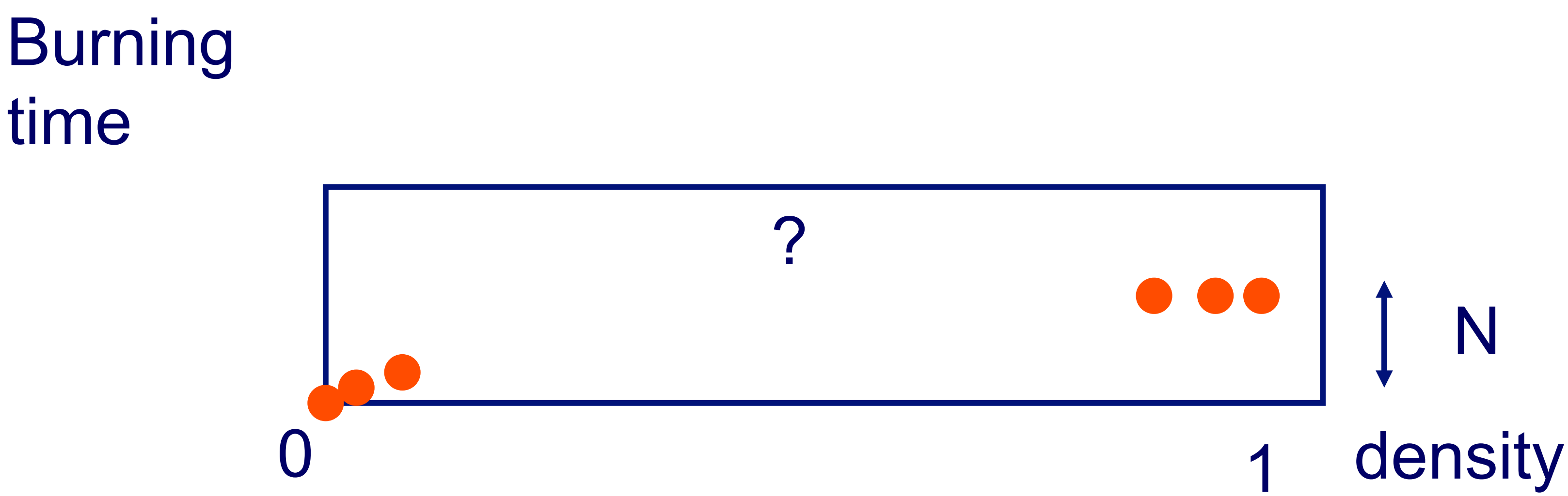


A burning tree will cause its neighbors to
 burn next.

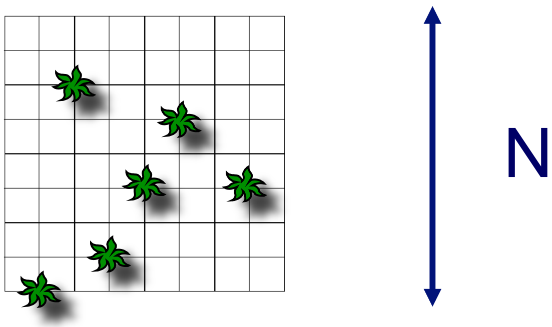
Which tree density *p* will cause the fire to last longest?

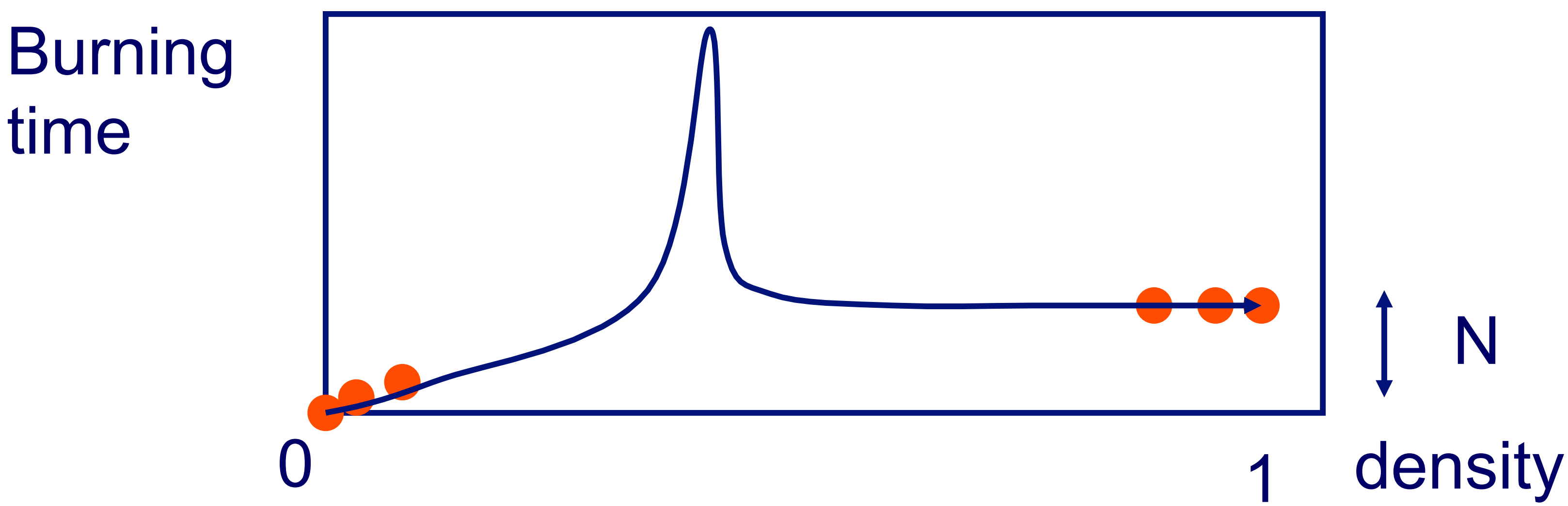
Percolation and forest fires



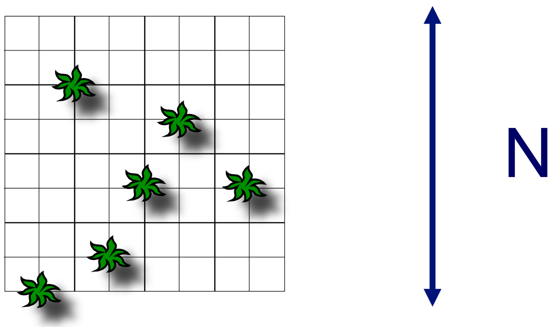


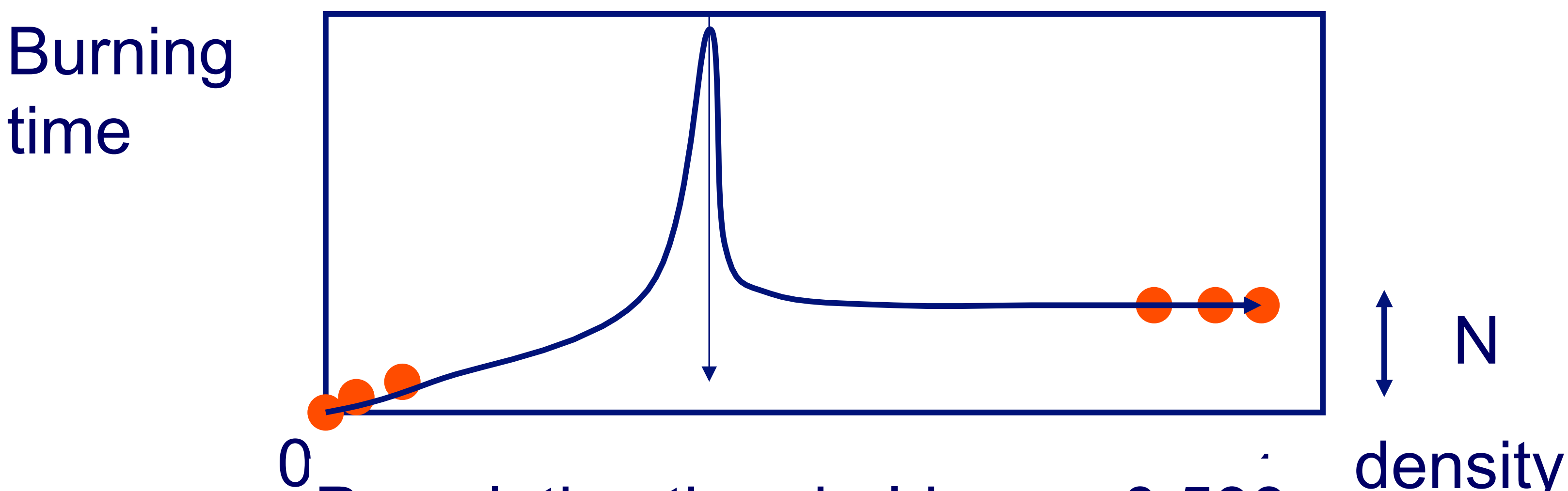
Percolation and forest fires



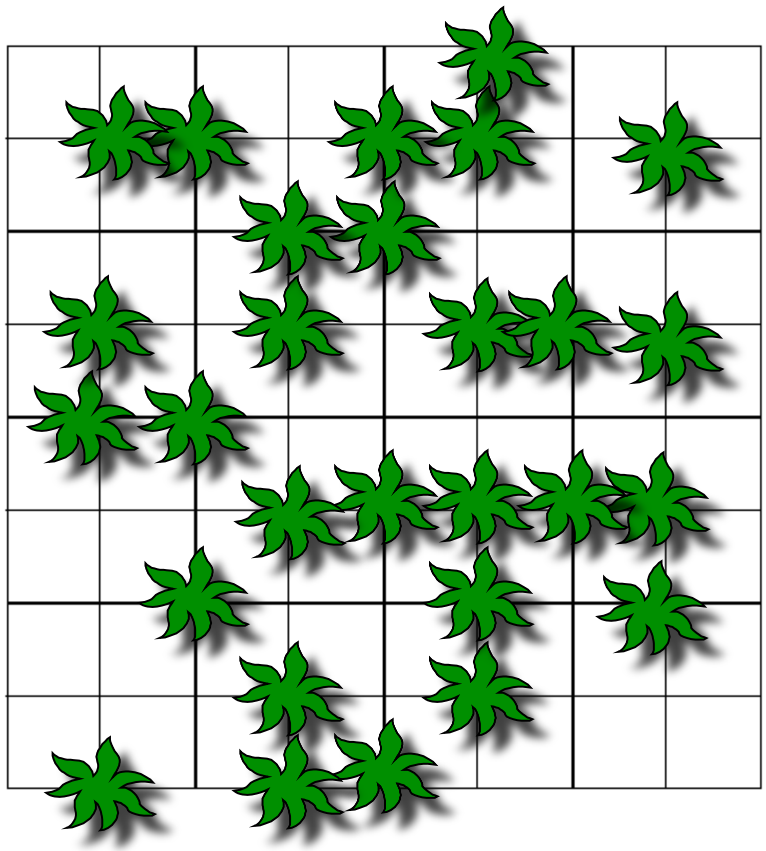


Percolation and forest fires





Percolation and forest fires



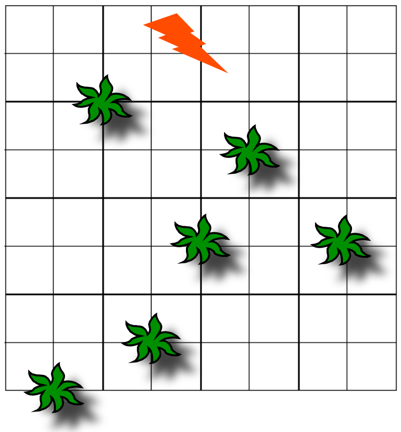
At pc ~ 0.593:
No characteristic scale;
'patches' of all sizes;
Korcak-like 'law'.

Outline

- Generative mechanisms
  - Random walk
  - Yule distribution = CRP
  - Percolation
  - Self-organized criticality
  - Other

Self-organized criticality

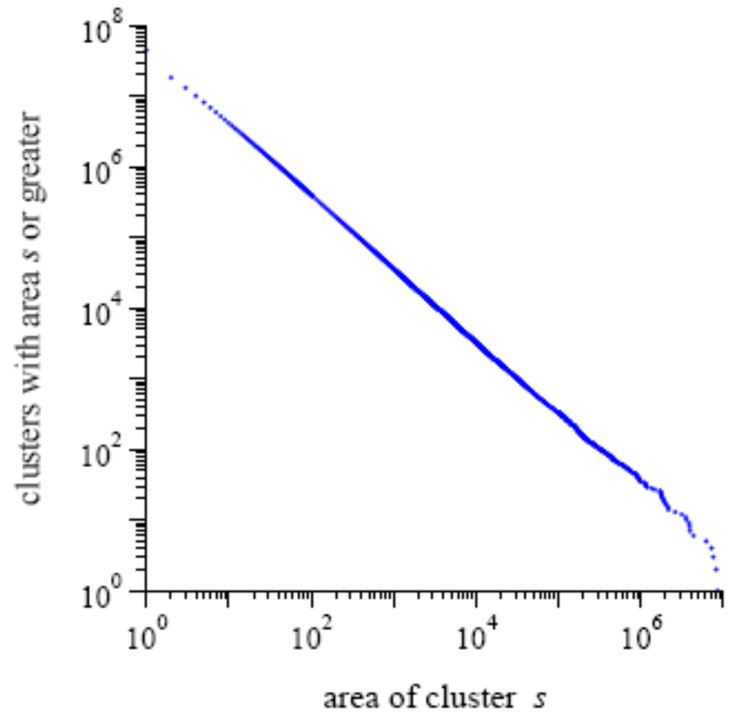
- Trees appear at random (eg., seeds, by the wind)
- Fires start at random (eg., lightning)
- Q1: What is the distribution of size of forest fires?



Self-organized criticality

- A1: Power law-like

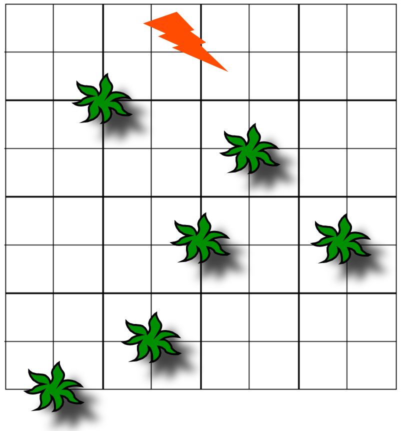
CCDF



Area of cluster s

Self-organized criticality

- Trees appear at random (eg., seeds, by the wind)
- Fires start at random (eg., lightning)
- Q2: what is the average density?



Self-organized criticality

- A2: the critical density pc ~ 0.593

Self-organized criticality

- [Bak]: size of avalanches ~ power law:
- Drop a grain randomly on a grid
- It causes an avalanche if height(x,y) is >1 higher than its four neighbors

[Per Bak: *How Nature works*, 1996]

Outline

- Generative mechanisms
  - Random walk
  - Yule distribution = CRP
  - Percolation
  - Self-organized criticality
  - Other

Other

- Random multiplication
- Fragmentation

-> lead to lognormals (~ look like power laws)

Others

Random multiplication:

- Start with C dollars; put in bank
- Random interest rate s(t) each year t
- Each year t: C(t) = C(t-1) * (1+ s(t))

- Log(C(t)) = log(C) + log(..) + log(..) … -> Gaussian

Others

Random multiplication:

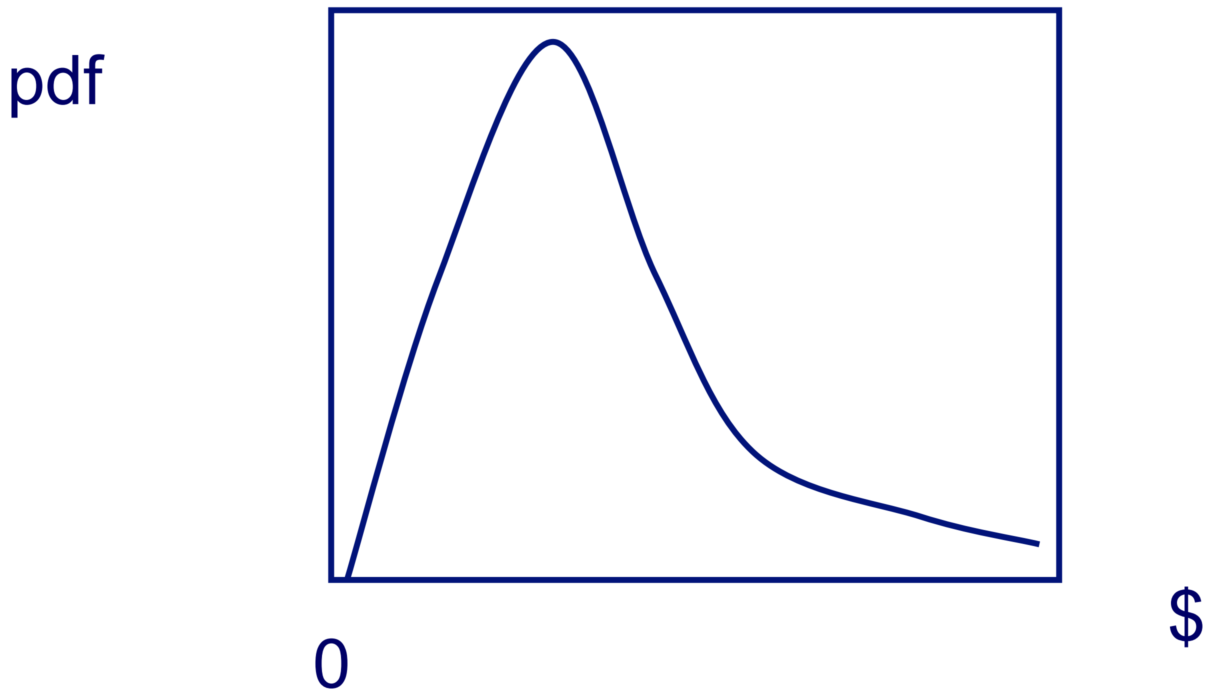
- Log(C(t)) = log(C) + log(..) + log(..) … -> Gaussian

- Thus C(t) = exp(Gaussian)
- By definition, this is Lognormal

Carnegie Mellon

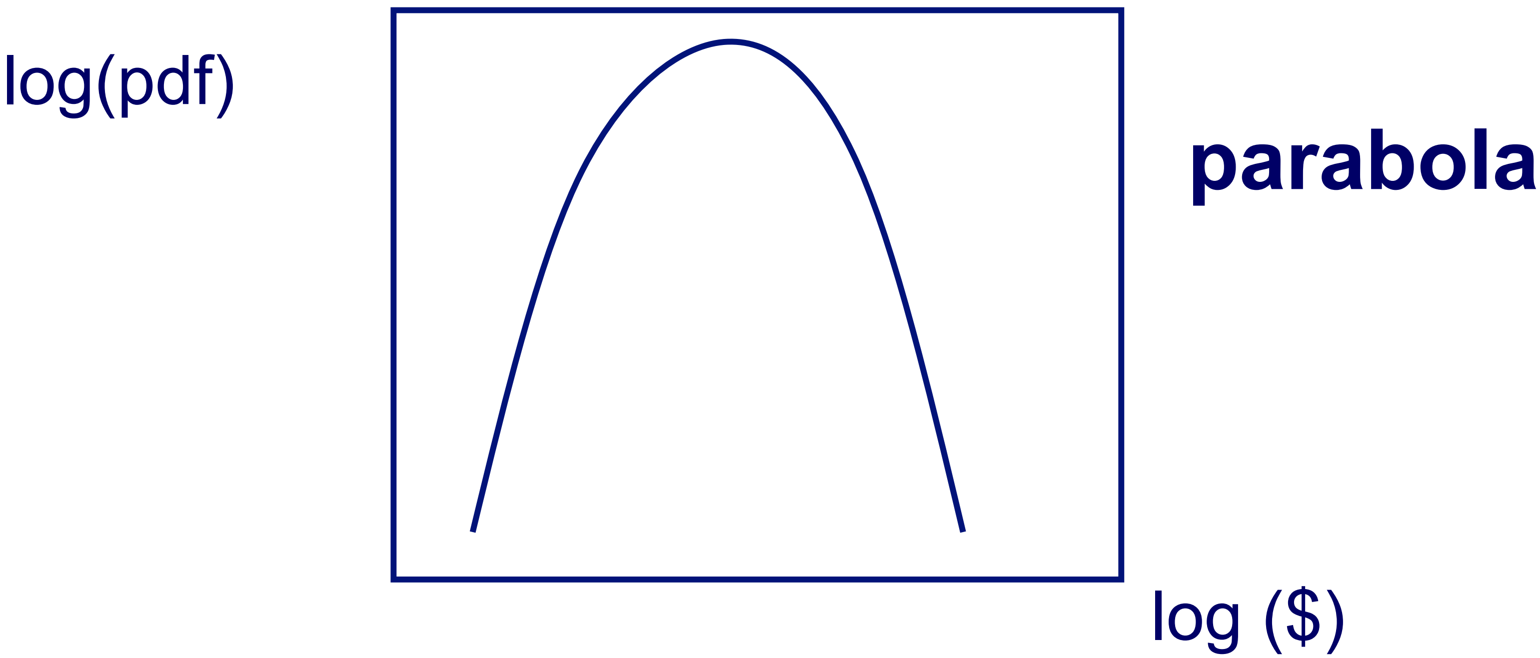
Others

Lognormal:



Others

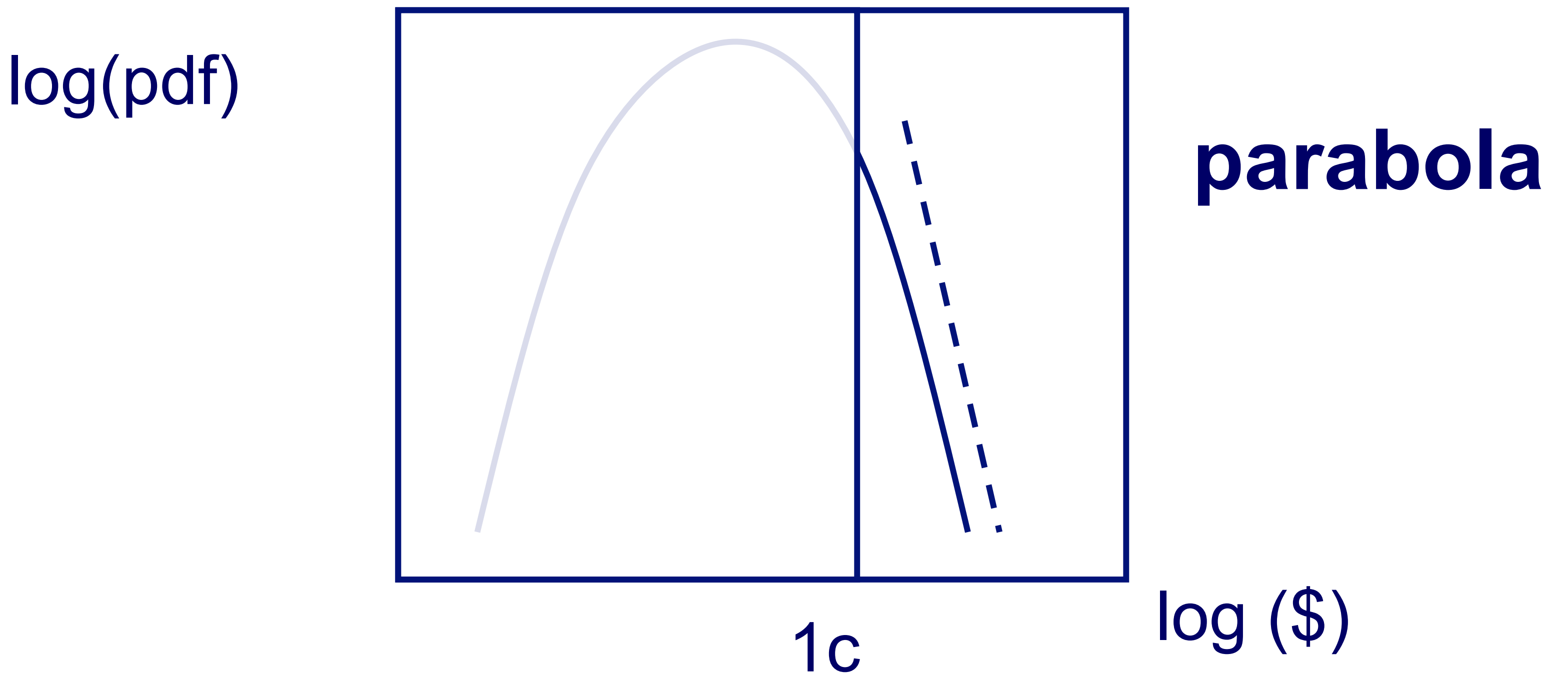
Lognormal:



Carnegie Mellon

Others

Lognormal:



Other

- Random multiplication
- Fragmentation

-> lead to lognormals (~ look like power laws)

Other

- Stick of length 1
- Break it at a random point x ($0<x<1$)
- Break each of the pieces at random

- Resulting distribution: lognormal (why?)

Conclusions

- Many, natural mechanisms, may yield power-laws (or log-normals etc)

Questions?

- www.cs.cmu.edu/~christos