Carnegie Mellon

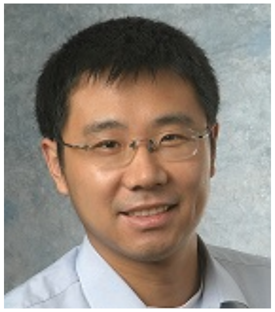
# Large Graph Mining - Patterns, Explanations and Cascade Analysis

*Christos Faloutsos*

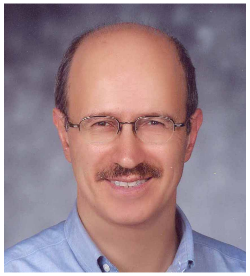
CMU

# Thank you!

- Prof. Xiang Zhang

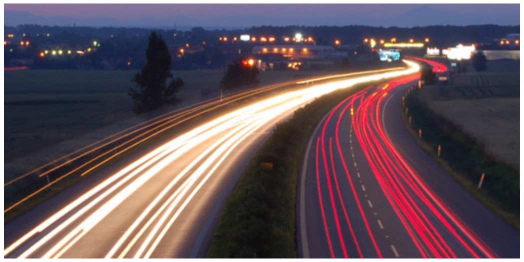


- Profs Meral & Tekin Ozsoyoglu



- Prof. Mike Lewicki

# Roadmap

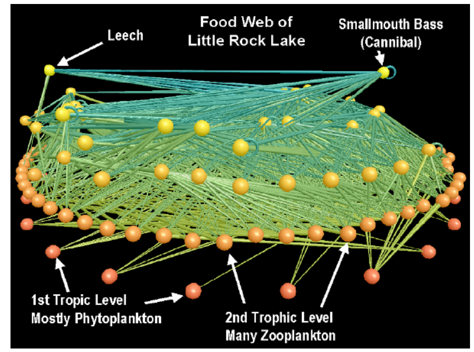


- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
- Conclusions

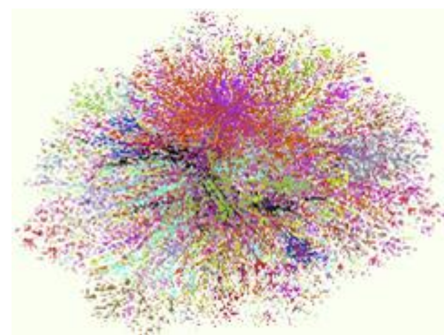
# Graphs - why should we care?

>$10B revenue

>0.5B users




Food Web
[Martinez '91]



Internet Map
[lumeta.com]

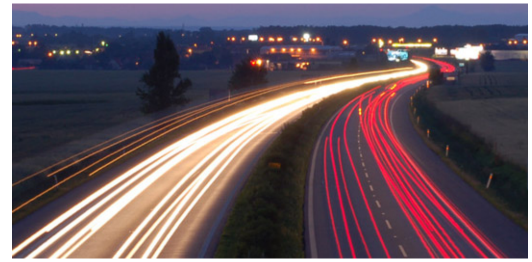
# Graphs - why should we care?

- web-log ('blog') news propagation YAHOO! BLOG
- computer network security: email/IP traffic and anomaly detection
- Recommendation systems NETFLIX
- ....

- Many-to-many db relationship -> graph

Carnegie Mellon

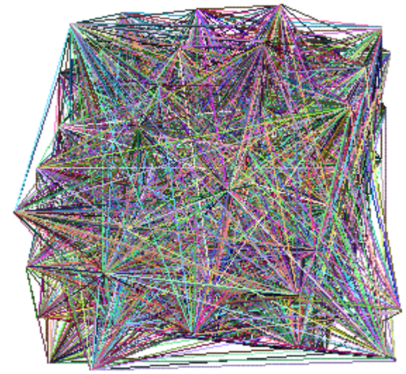
# Roadmap



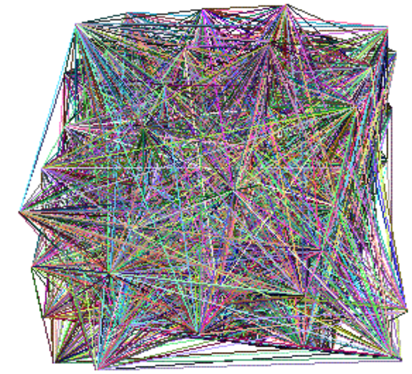
- Introduction – Motivation
- ➡ Part#1: Patterns in graphs
  - Static graphs
  - Time-evolving graphs
  - Why so many power-laws?
- Part#2: Cascade analysis
- Conclusions

# Part 1: Patterns & Laws

# Laws and patterns



- Q1: Are real graphs random?



# Laws and patterns

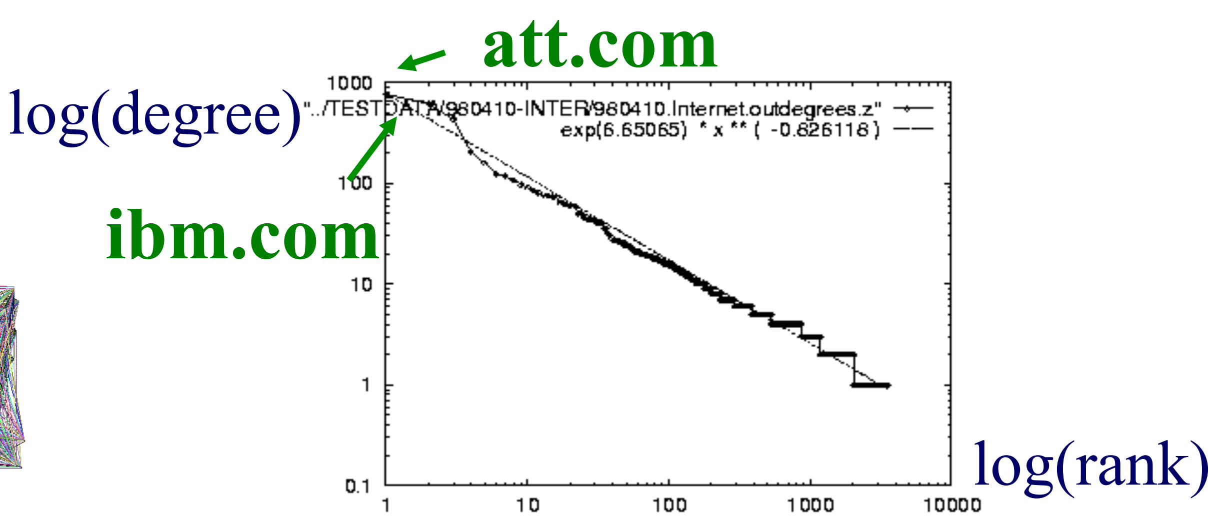
- Q1: Are real graphs random?
- A1: NO!!
  – Diameter
  – in- and out- degree distributions
  – other (surprising) patterns
- Q2: why ‘no good cuts’?
- A2: <self-similarity – stay tuned>

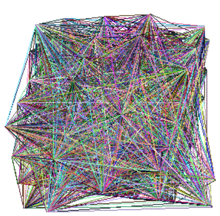
- So, let’s look at the data

# Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

**internet domains**

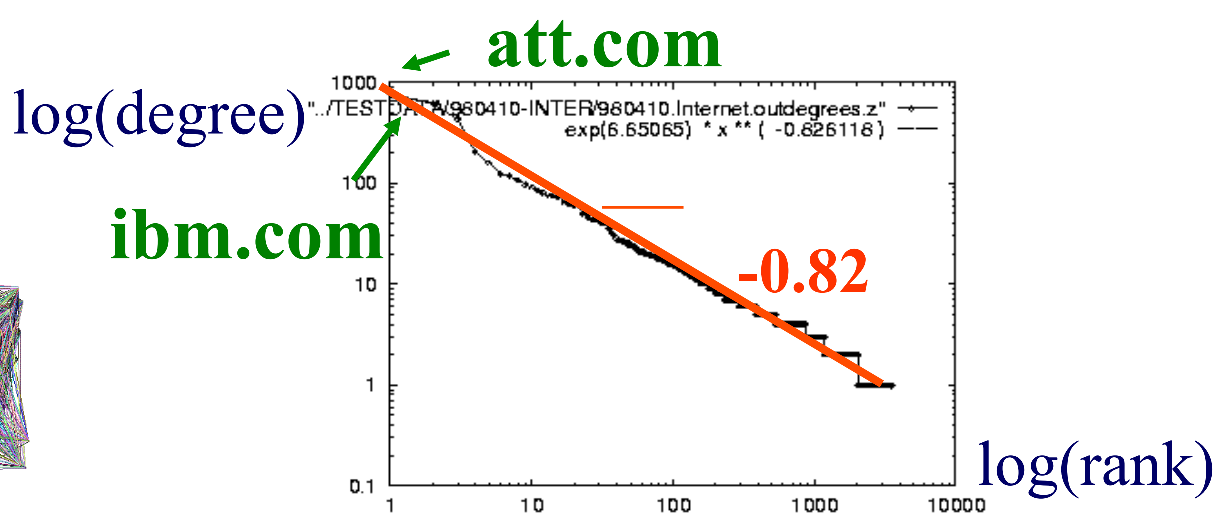


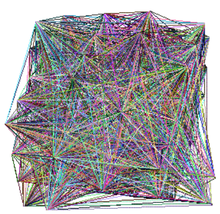



# Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

**internet domains**



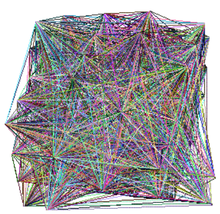



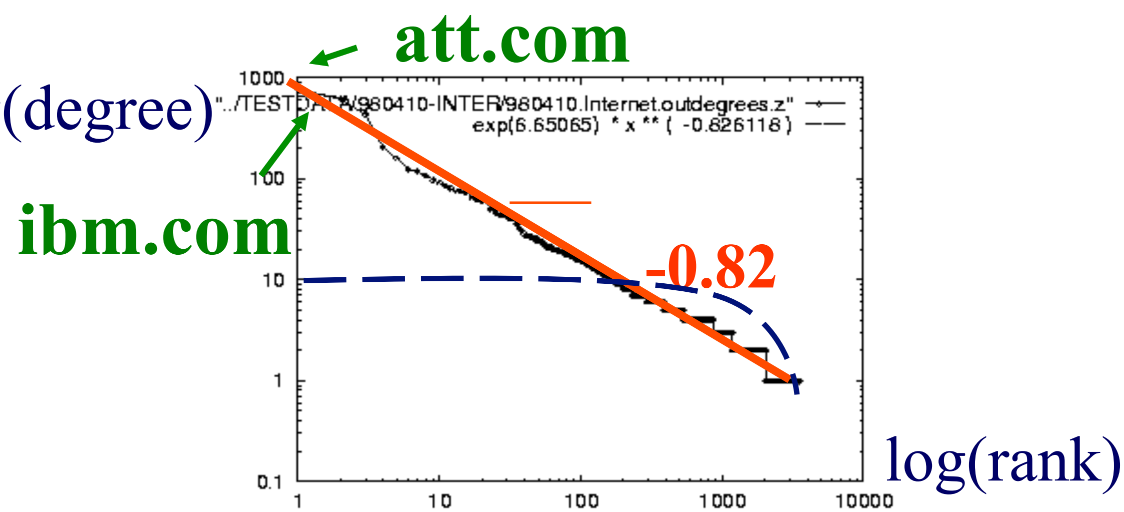
Carnegie Mellon

# Solution# S.1

- Q: So what?

**internet domains**





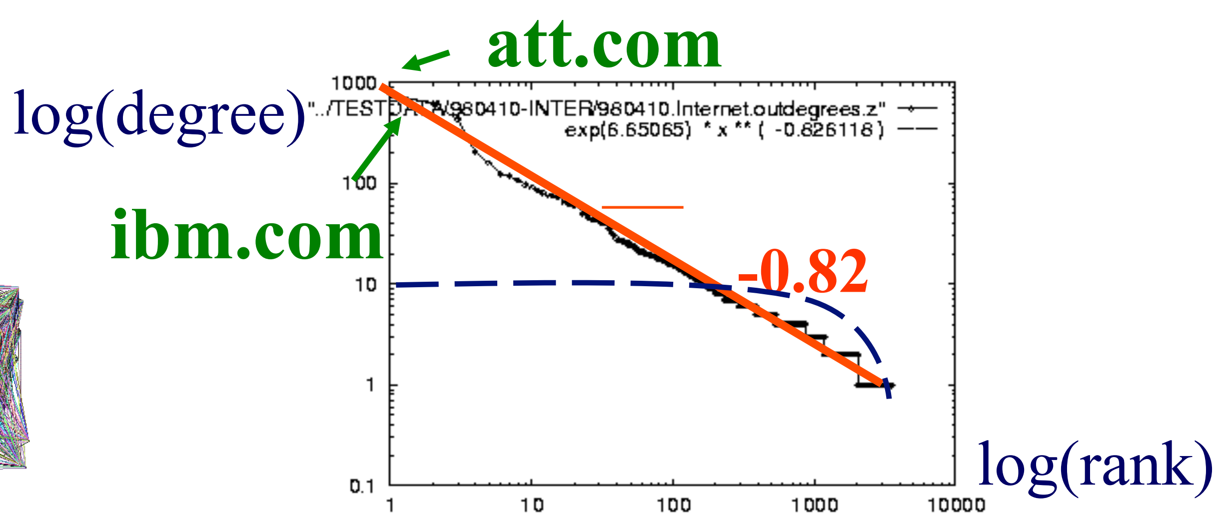
# Solution# S.1

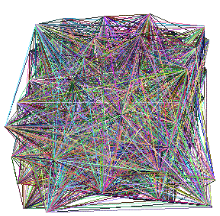
- Q: So what?

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

- A1: # of two-step-away pairs:

**internet domains**



Gaussian trap

# Solution# S.1

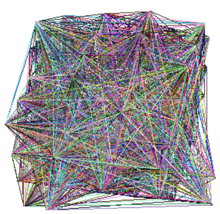
- Q: So what?
- A1: # of two-step-away pairs: O(d_max ^2) ~ 10M^2

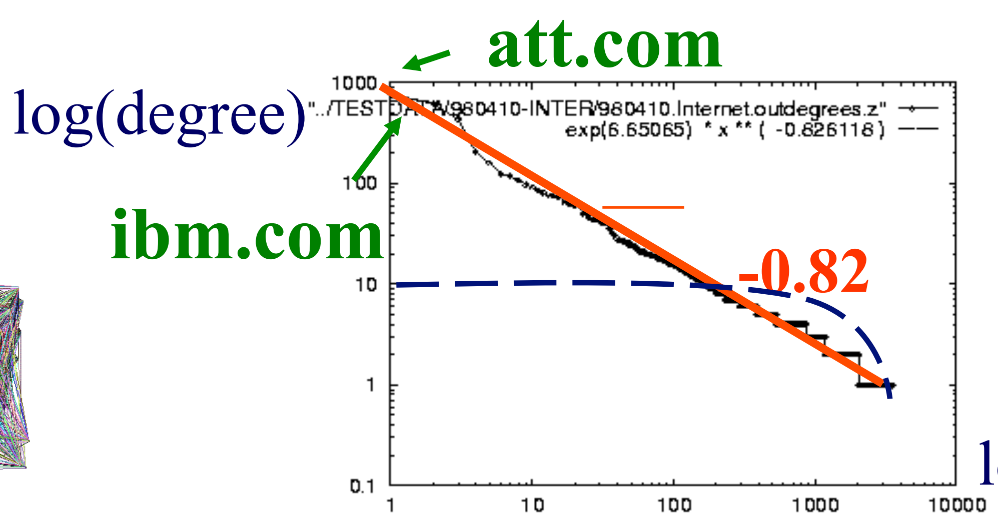
= friends of friends (F.O.F.)

⇩

~0.8PB ->
a data center(!)

**internet domains**






DCO @ CMU

Gaussian trap

# Solution# S.1

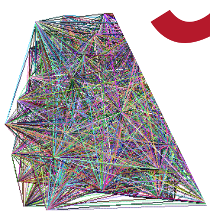
- Q: So what?
- A1: # of two-step-aw ?) ~ 10M^2

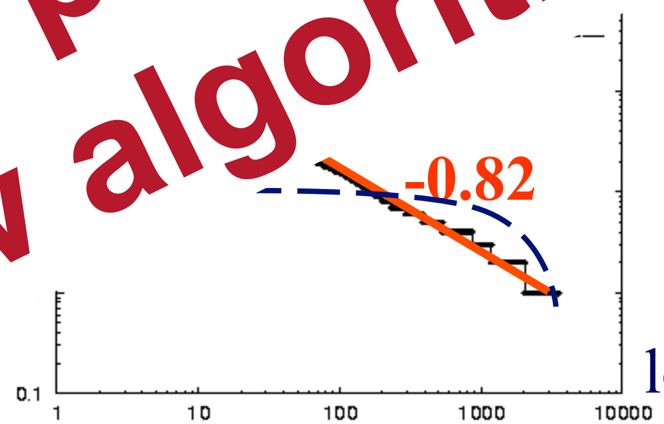
**inte**

⇩

~0.8PB ->
a data center(!)

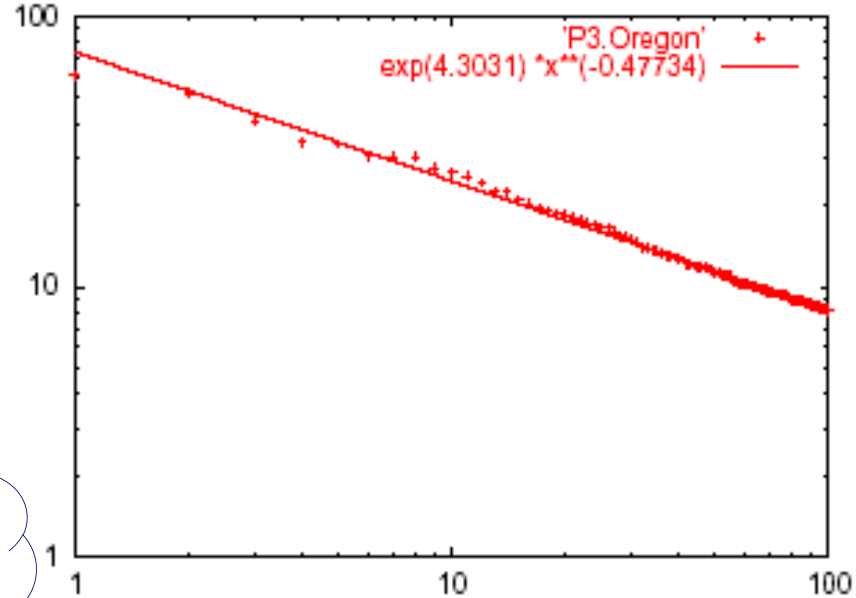
Such patterns ->
New algorithms






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

Eigenvalue




Rank of decreasing eigenvalue

Exponent = slope

$E = -0.48$

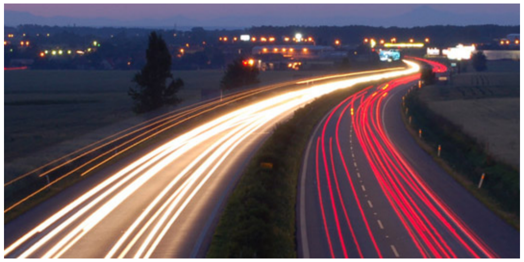
May 2001

$\mathbf{A\,x} = \lambda\,\mathbf{x}$

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

Carnegie Mellon

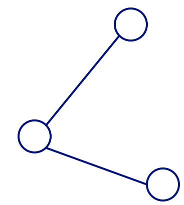
# Roadmap



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

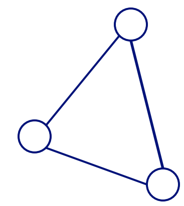
Carnegie Mellon

# Solution# S.3: Triangle 'Laws'



- Real social networks have a lot of triangles

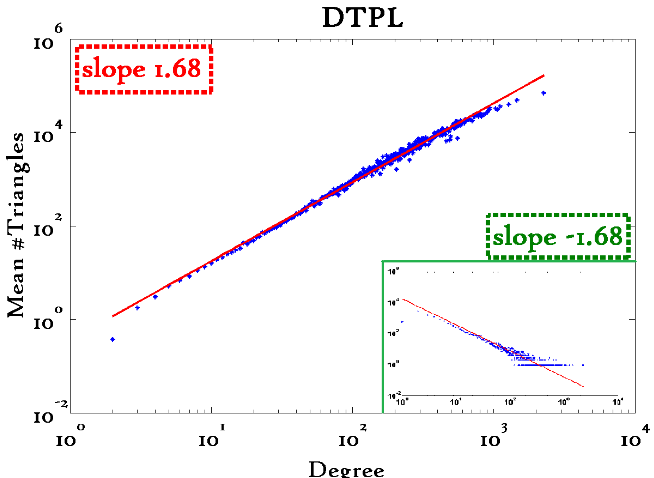
# Solution# S.3: Triangle 'Laws'



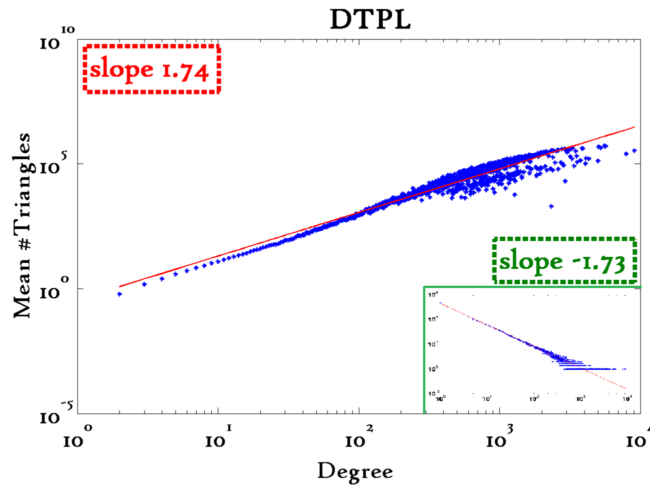
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?

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

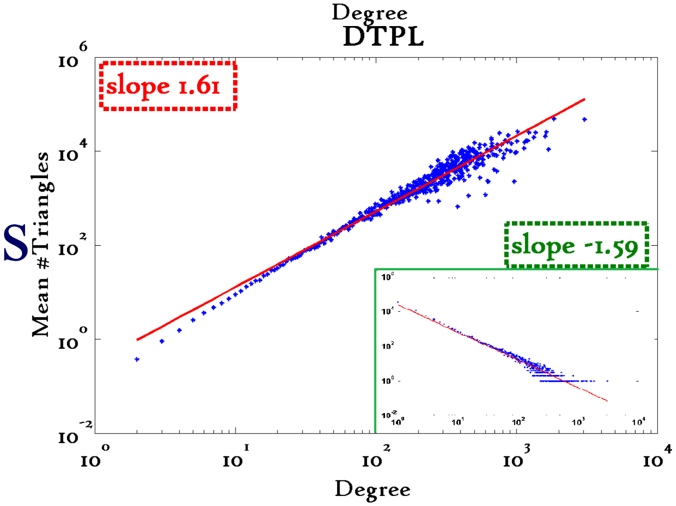
Reuters

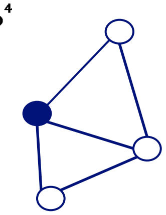



SN




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) – $O(d_{max}^2)$
Q: Can we do that quickly?
A:

details

# Triangle Law: Computations
## [Tsourakakis ICDM 2008]

But: triangles are expensive to compute

(3-way join; several approx. algos) – $O(d_{max}^2)$

Q: Can we do that quickly?

A: Yes!

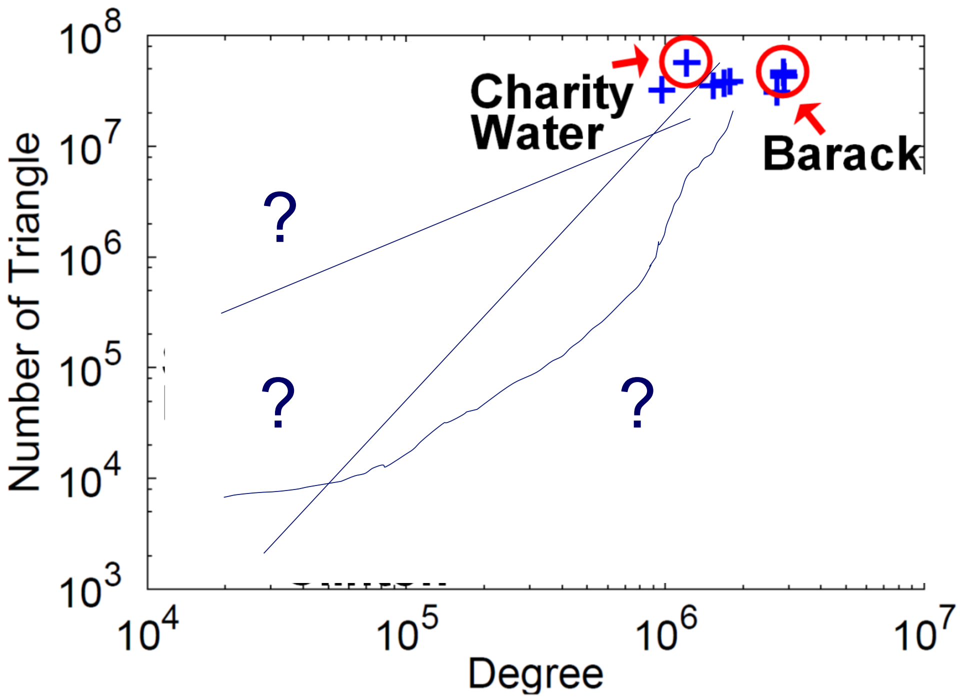
**#triangles = 1/6 Sum ( $\lambda_i^3$ )**

$A x = \lambda x$

(and, because of skewness (S2) ,

we only need the top few eigenvalues! - O(E)

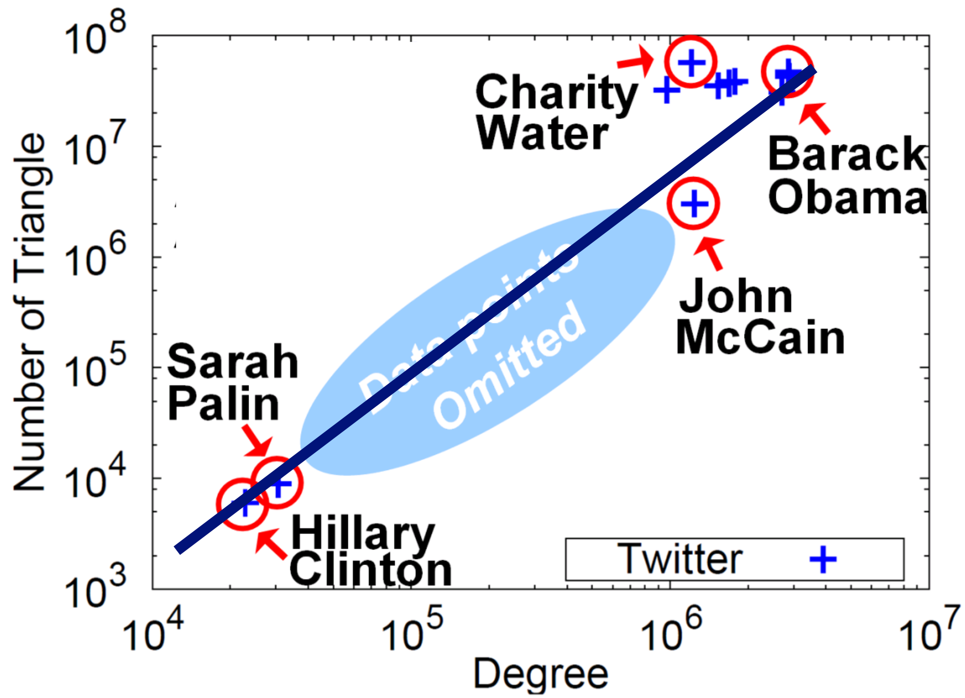
# Triangle counting for large graphs?


Anomalous nodes in Twitter(~ 3 billion edges)

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

# Triangle counting for large graphs?

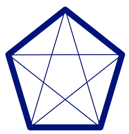



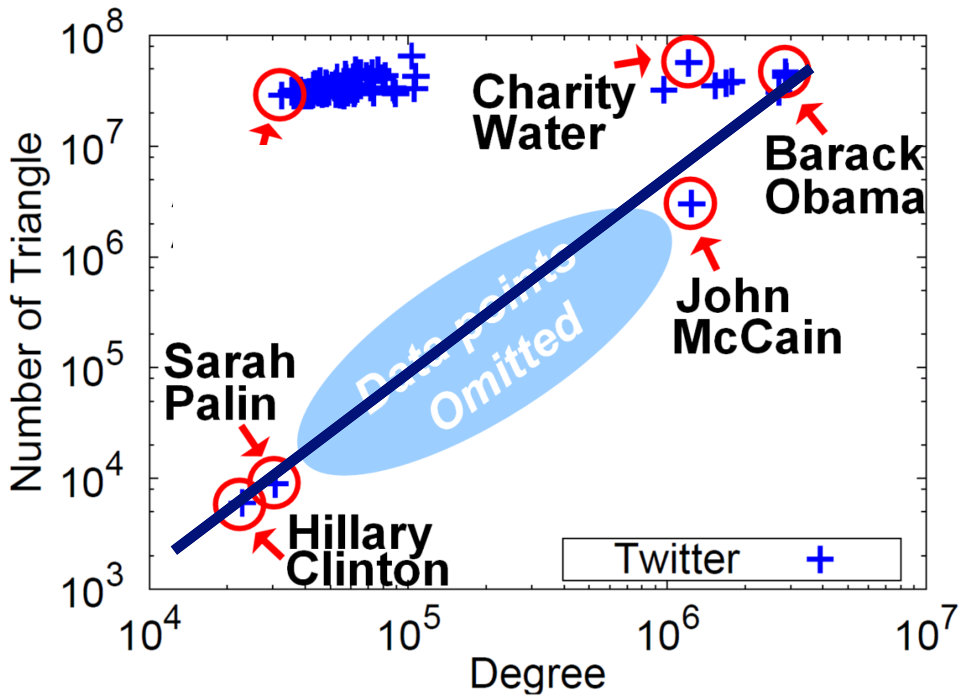

Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

# Triangle counting for large graphs?



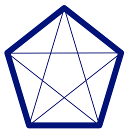


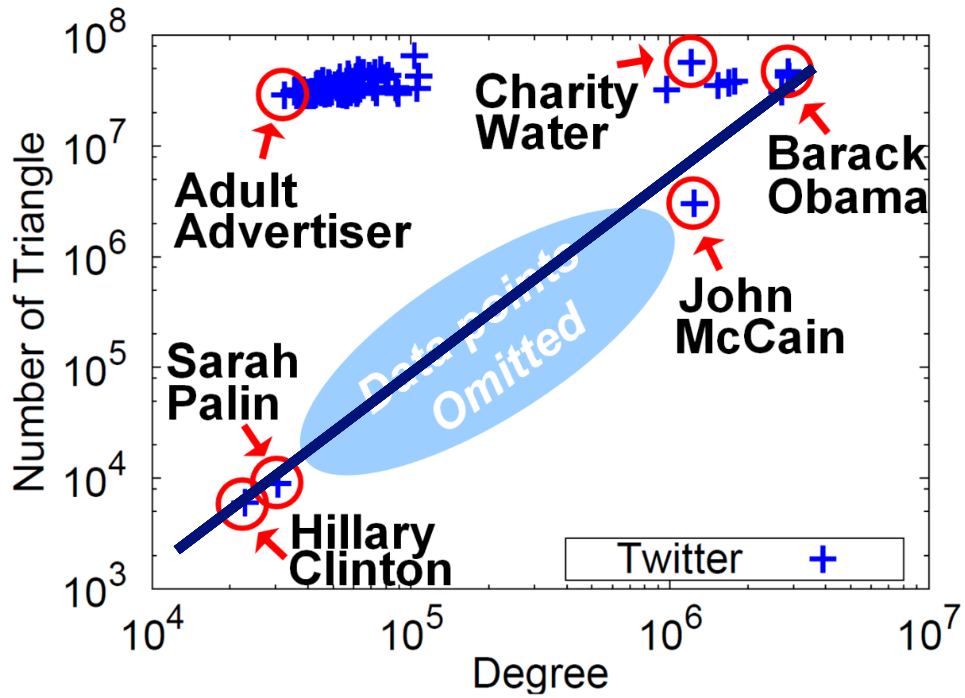


Anomalous nodes in Twitter(~ 3 billion edges)

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

# Triangle counting for large graphs?





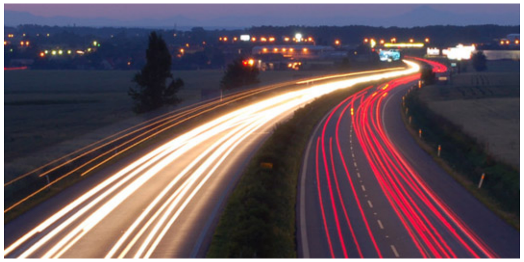


Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

Carnegie Mellon

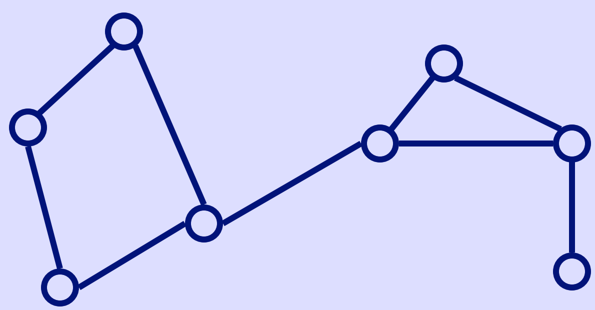
# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - Static graphs
    - Power law degrees; eigenvalues; triangles
    - ➡ Anti-pattern: NO good cuts!
  - Time-evolving graphs
- ….
- Conclusions

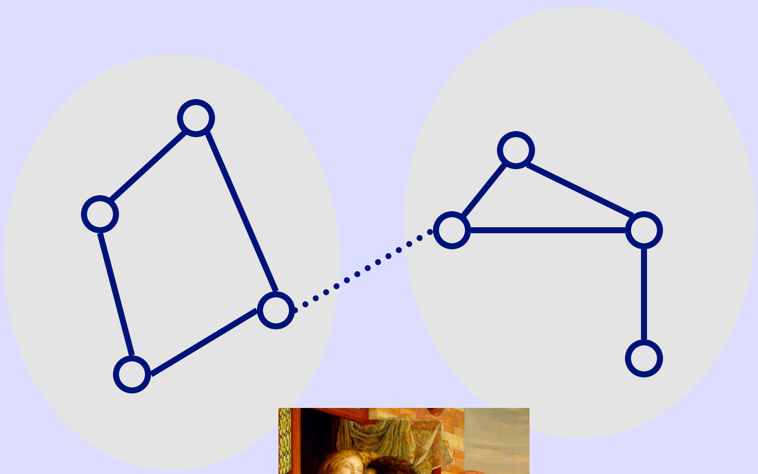
# Background: Graph cut problem

- Given a graph, and *k*
- Break it into *k* (disjoint) communities

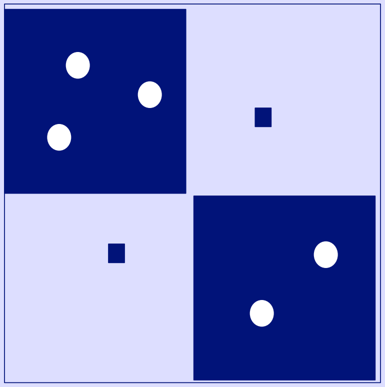


# Graph cut problem

- Given a graph, and *k*
- Break it into *k* (disjoint) communities
- (assume: block diagonal = 'cavemen' graph)



$k = 2$



# Many algo's for graph partitioning

- METIS [Karypis, Kumar +]
- 2nd eigenvector of Laplacian
- Modularity-based [Girwan+Newman]
- Max flow [Flake+]
- …
- …
- …

Carnegie Mellon

# Strange behavior of min cuts

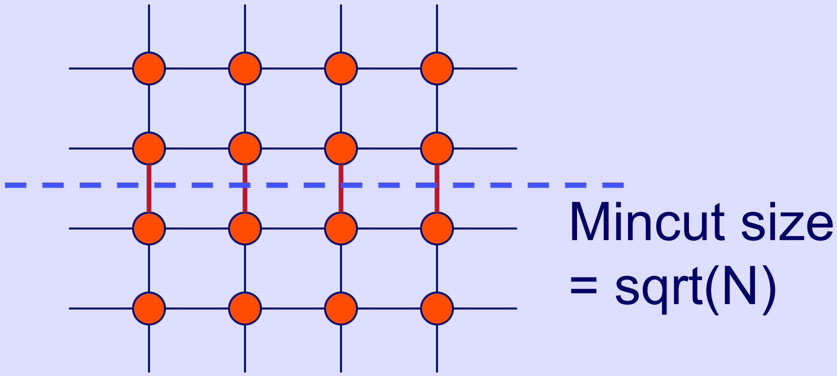
- Subtle details: next
  - Preliminaries: min-cut plots of 'usual' graphs

*NetMine: New Mining Tools for Large Graphs*, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

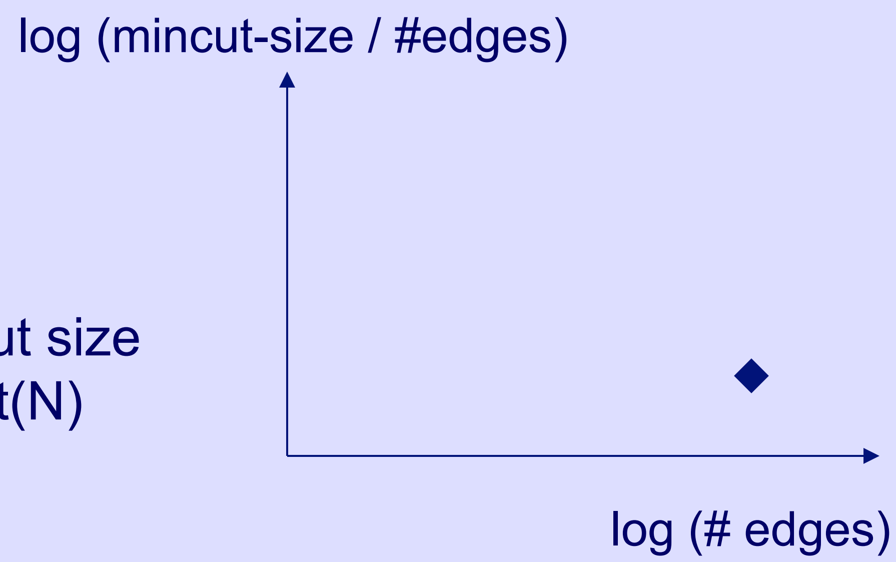
*Statistical Properties of Community Structure in Large Social and Information Networks, J.* Leskovec, K. Lang, A. Dasgupta, M. Mahoney. WWW 2008.

# "Min-cut" plot

- Do min-cuts recursively.



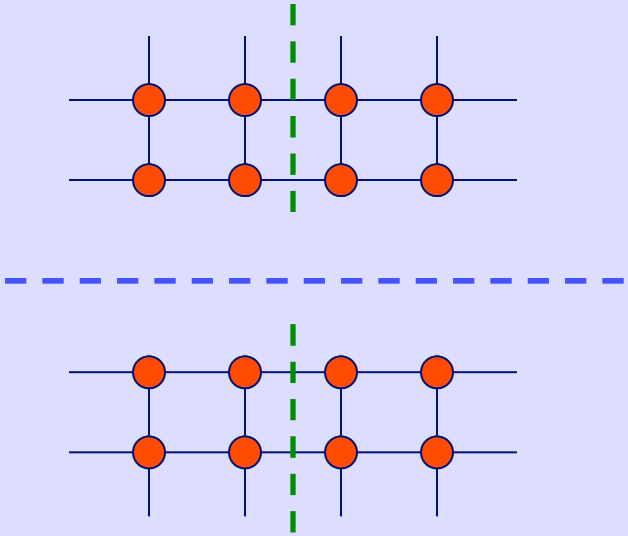

N nodes



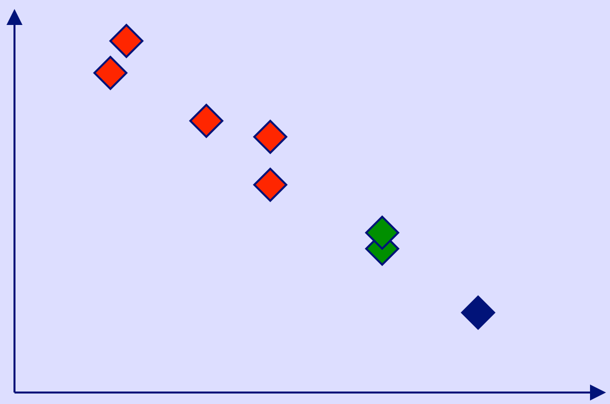
# "Min-cut" plot

- Do min-cuts recursively.

New min-cut

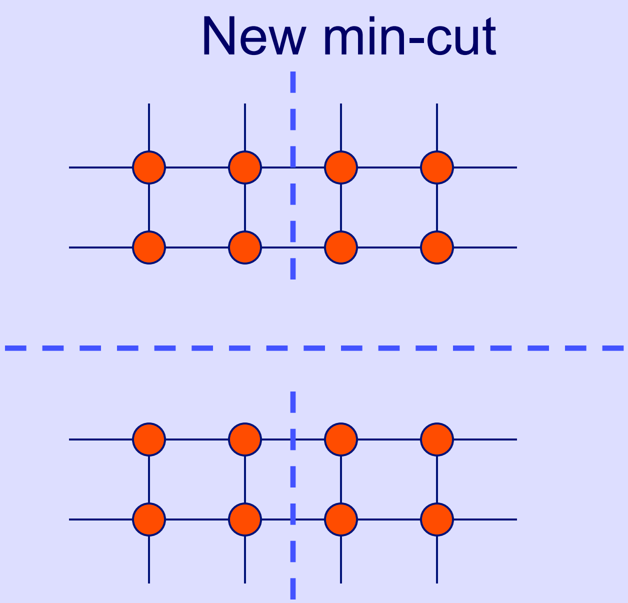


N nodes

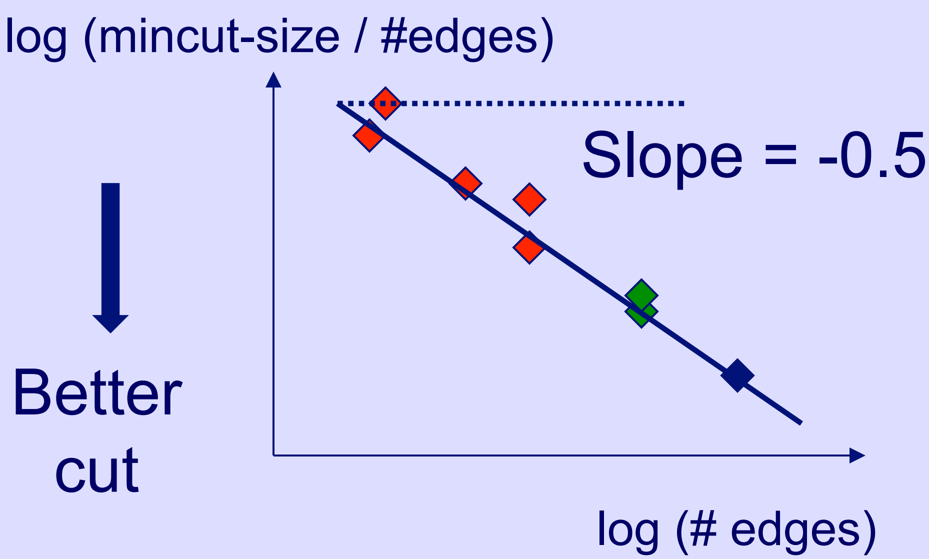


# "Min-cut" plot

- Do min-cuts recursively.

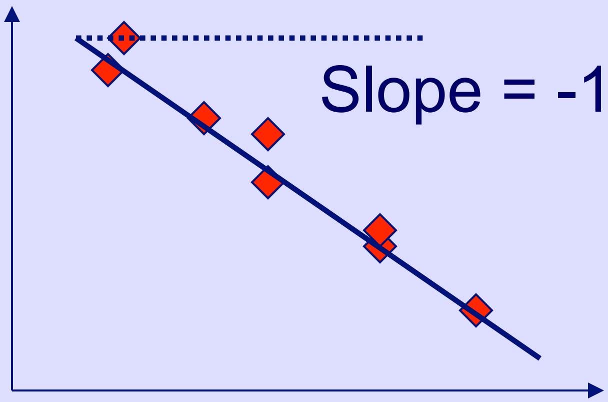


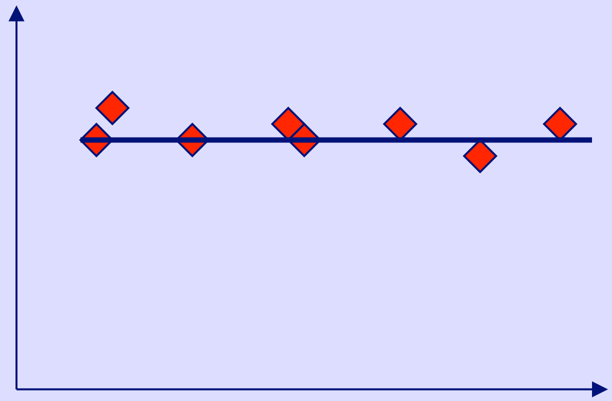

Better cut



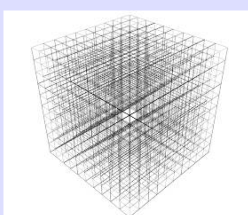

N nodes

# “Min-cut” plot




For a d-dimensional grid, the slope is -1/d



For a random graph (and clique), the slope is 0

Carnegie Mellon

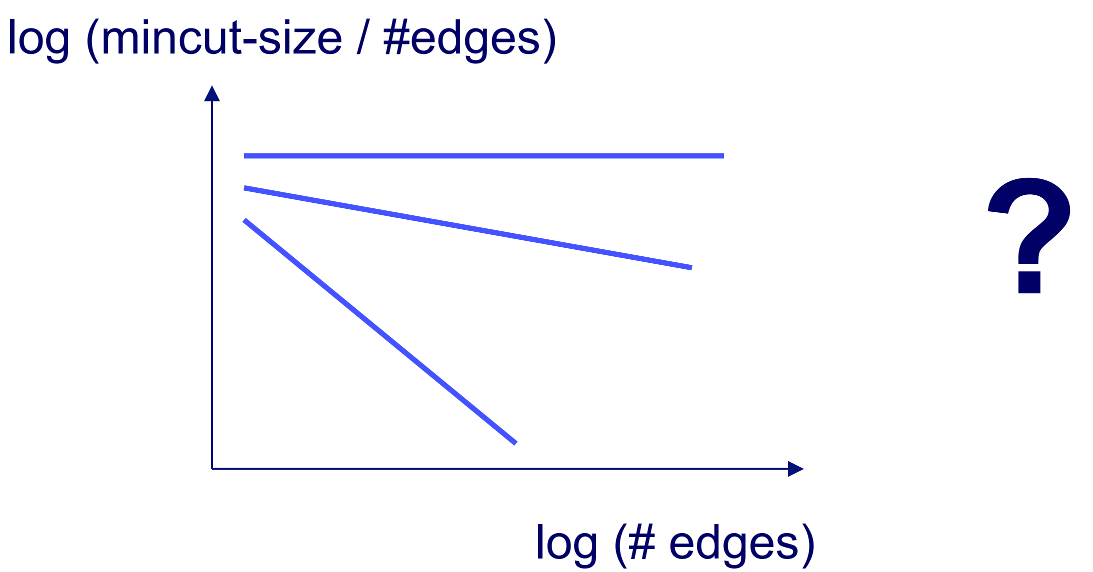
# Experiments

- Datasets:
  - Google Web Graph: 916,428 nodes and 5,105,039 edges
  - Lucent Router Graph: Undirected graph of network routers from www.isi.edu/scan/mercator/maps.html; 112,969 nodes and 181,639 edges
  - User ➔ Website Clickstream Graph: 222,704 nodes and 952,580 edges

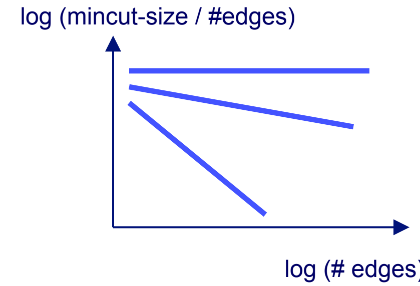
*NetMine: New Mining Tools for Large Graphs*, by D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch, in the SDM 2004 Workshop on Link Analysis, Counter-terrorism and Privacy

# "Min-cut" plot

- What does it look like for a real-world graph?

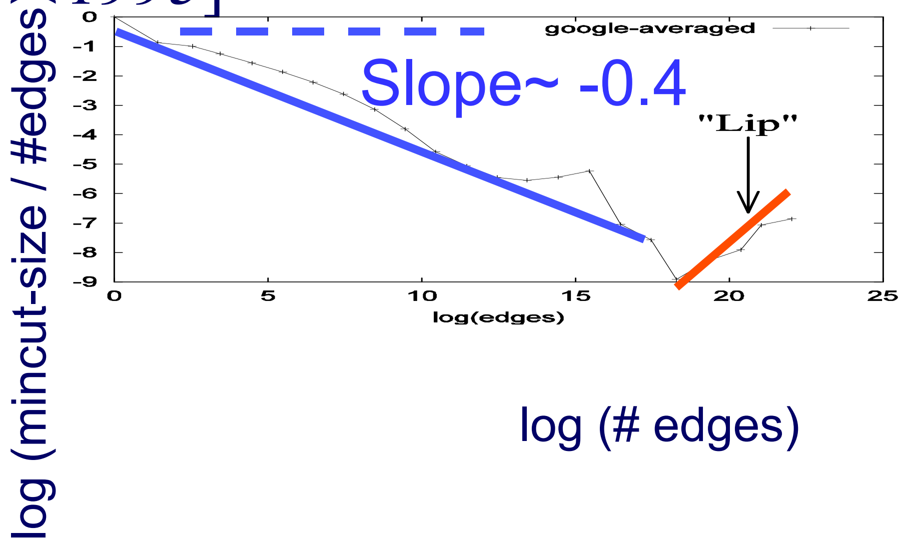


Carnegie Mellon

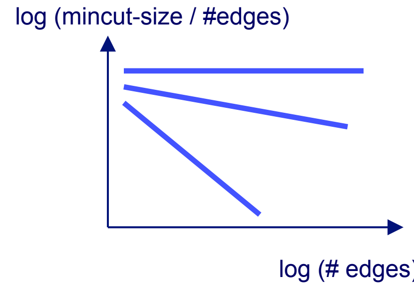



# Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]

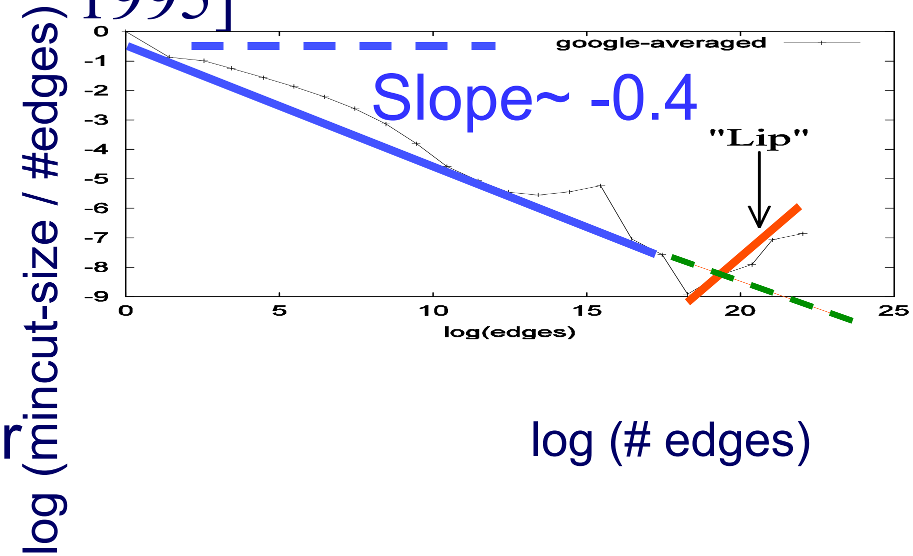



- Google Web graph
- Values along the y-axis are averaged
- "lip" for large # edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!




# Experiments

- Used the METIS algorithm [Karypis, Kumar, 1995]



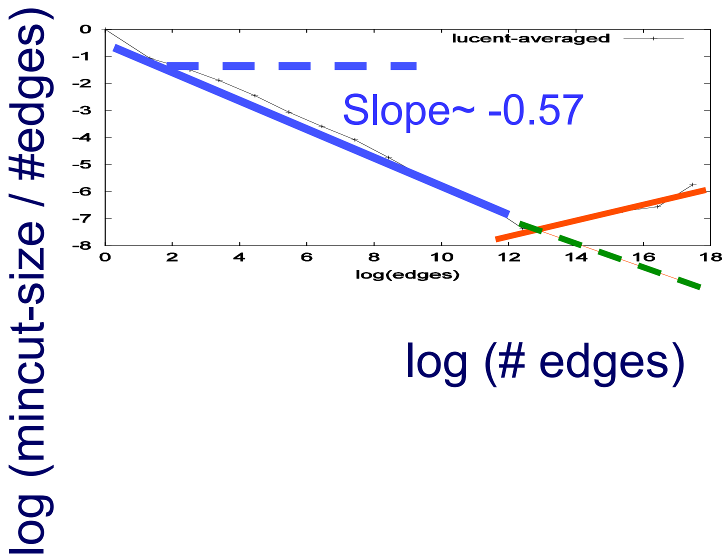

Better cut

- Google Web graph
- Values along the y-axis are averaged
- “lip” for large # edges
- Slope of -0.4, corresponds to a 2.5-dimensional grid!

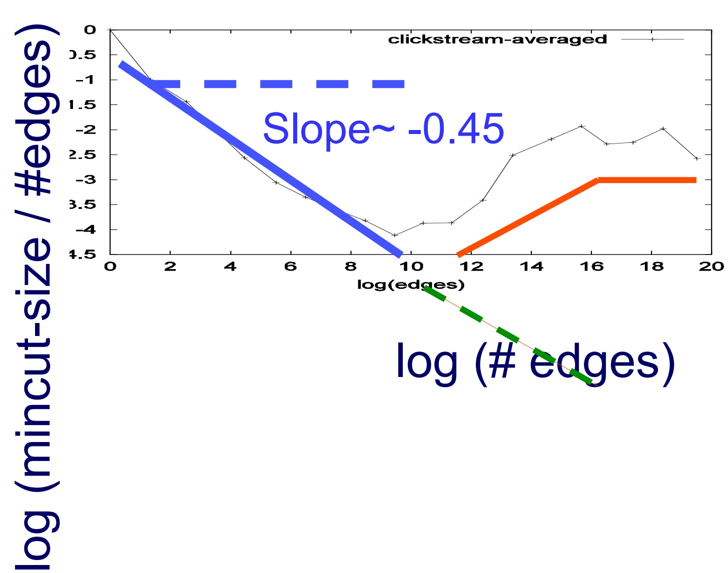
Carnegie Mellon

# Experiments

- Same results for other graphs too…




Lucent Router graph

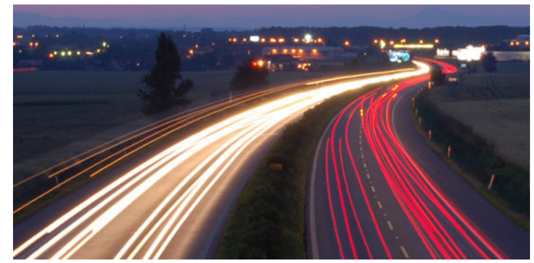



Clickstream graph

# Why no good cuts?

- Answer: self-similarity (few foils later)

# Roadmap



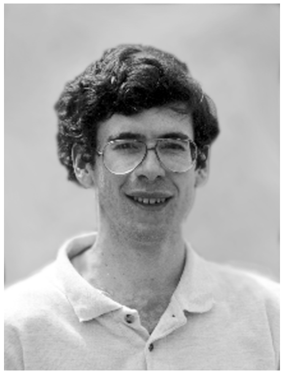
- Introduction – Motivation
- Part#1: Patterns in graphs
  - Static graphs
  - ➡ Time-evolving graphs
  - Why so many power-laws?
- Part#2: Cascade analysis
- Conclusions

Carnegie Mellon

# Problem: Time evolution

- with Jure Leskovec (CMU -> Stanford)

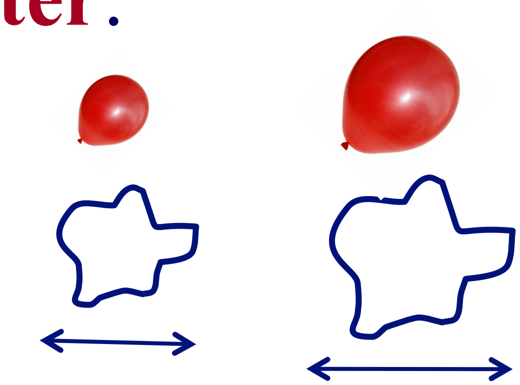
- and Jon Kleinberg (Cornell – sabb. @ CMU)

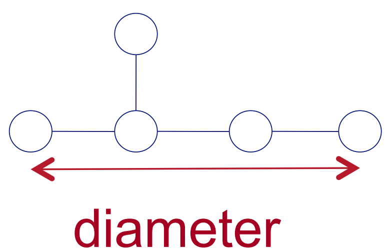


Jure Leskovec, Jon Kleinberg and Christos Faloutsos: *Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations*, KDD 2005

# T.1 Evolution of the Diameter

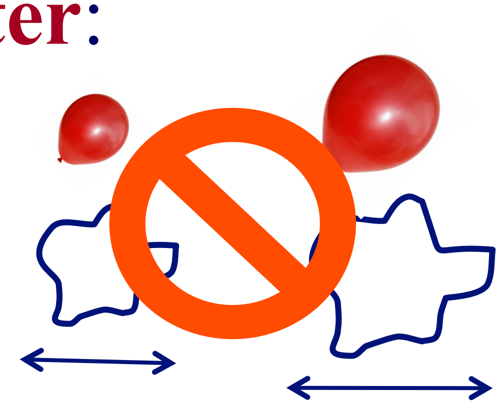
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - [diameter ~ O( $N^{1/3}$)]
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?





# T.1 Evolution of the Diameter

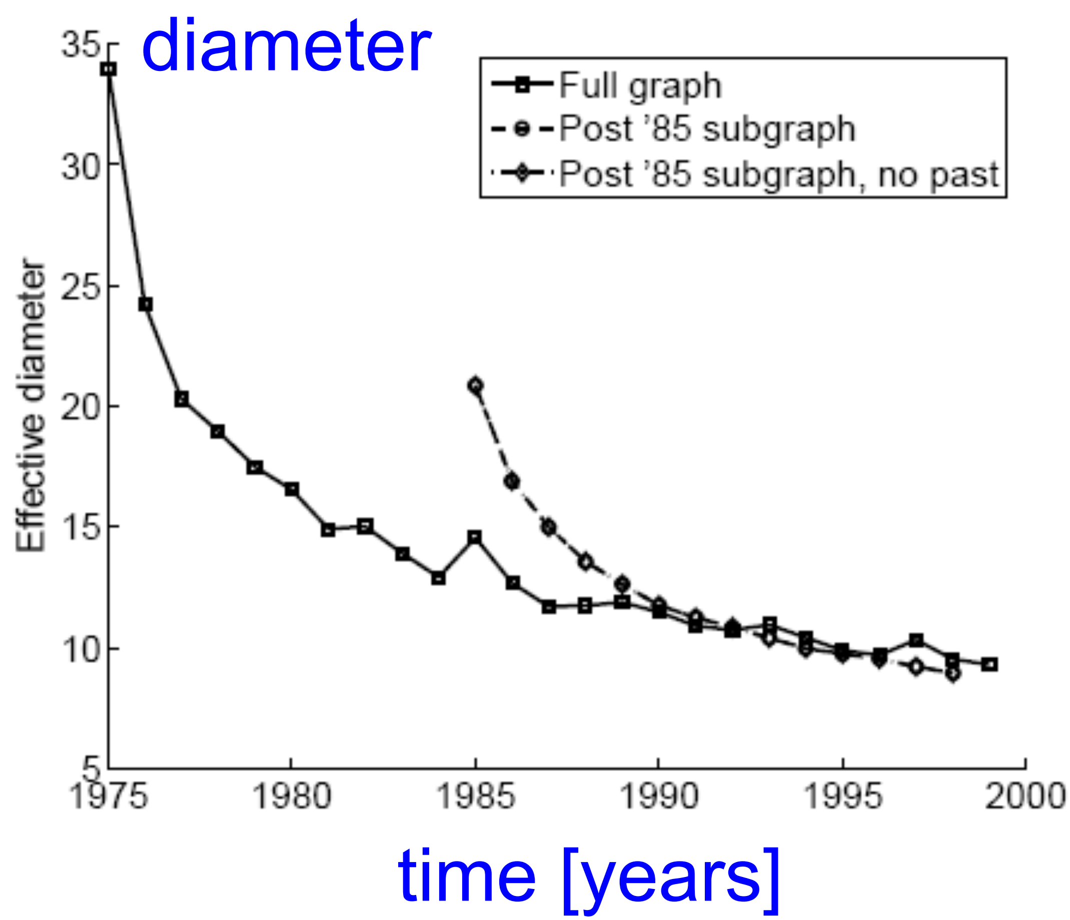
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - [diameter ~ $O(N^{1/3})$]
  - diameter ~ O(log N)
  - diameter ~ O(log log N)



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

# 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

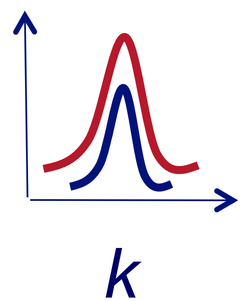
- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

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

- Q: what is your guess for

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

Say, *k* friends on average



# T.2 Temporal Evolution of the Graphs

**Gaussian trap**

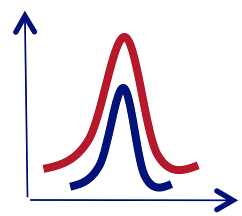
- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

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

Say, *k* friends on average

- Q: what is your guess for

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



- A: over-doubled! ~ 3x
  - But obeying the ``Densification Power Law''

# T.2 Temporal Evolution of the Graphs

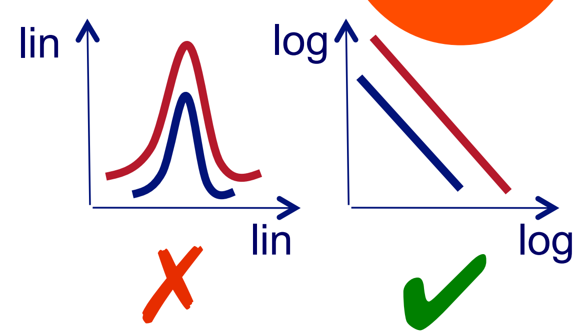
- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

  N(t+1) = 2 * N(t)
- Q: what is your guess for

  E(t+1) =?= 2 * E(t)
- A: over-doubled! ~ 3x
  - But obeying the ``Densification Power Law''

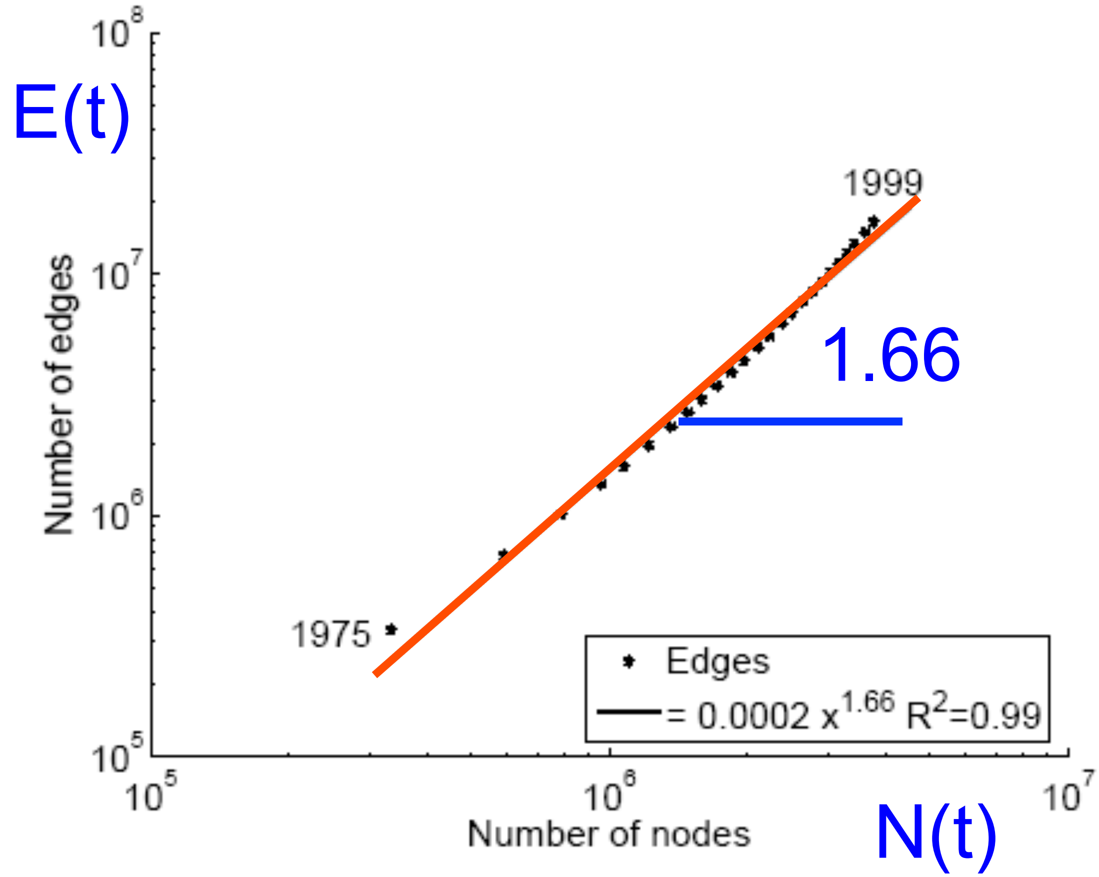
**Gaussian trap**

Say, *k* friends on average



# T.2 Densification – Patent Citations

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



Carnegie Mellon

# MORE Graph Patterns

| | Unweighted | Weighted |
|---|---|---|
| Static | **L01.** Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04]<br>**L02.** Triangle Power Law (TPL) [Tsourakakis `08]<br>**L03.** Eigenvalue Power Law (EPL) [Siganos et al. `03]<br>**L04.** Community structure [Flake et al. `02, Girvan and Newman `02] | **L10.** Snapshot Power Law (SPL) [McGlohon et al. `08] |
| Dynamic | **L05.** Densification Power Law (DPL) [Leskovec et al. `05]<br>**L06.** Small and shrinking diameter [Albert and Barabási `99, Leskovec et al. `05]<br>**L07.** Constant size 2nd and 3rd connected components [McGlohon et al. `08]<br>**L08.** Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. `08]<br>**L09.** Bursty/self-similar edge/weight additions [Gomez and Santonja `98, Gribble et al. `98, Crovella and | **L11.** Weight Power Law (WPL) [McGlohon et al. `08] |

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

# MORE Graph Patterns

| | Unweighted | Weighted |
|---|---|---|
| Static | L01. Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04]<br>L02. Triangle Power Law (TPL) [Tsourakakis `08]<br>L03. Eigenvalue Power Law (EPL) [Siganos et al. `03]<br>L04. Community structure [Flake et al. `02, Girvan and Newman `02] | L10. Snapshot Power Law (SPL) [McGlohon et al. `08] |
| Dynamic | L05. Densification Power Law (DPL) [Leskovec et al. `05]<br>L06. Small and shrinking diameter [Albert and Barabási `99, Leskovec et al. `05]<br>L07. Constant size 2nd and 3rd connected components [McGlohon et al. `08]<br>L08. Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. `08]<br>L09. Bursty/self-similar edge/weight additions [Gomez and Santonja `98, Gribble et al. `98, Crovella and | L11. Weight Power Law (WPL) [McGlohon et al. `08] |

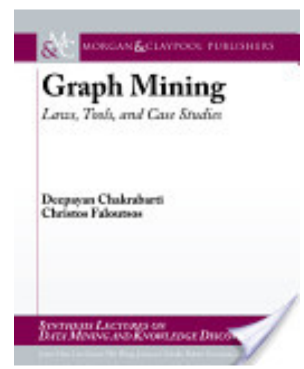
*RTG: A Recursive Realistic Graph Generator using Random Typing* Leman Akoglu and Christos Faloutsos. *PKDD*'09.

Carnegie Mellon

# MORE Graph Patterns

| | Unweighted | Weighted |
|---|---|---|
| Static | L01. Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04]<br>L02. Triangle Power Law (TPL) [Tsourakakis `08]<br>L03. Eigenvalue Power Law (EPL) [Siganos et al. `03]<br>L04. Community structure [Flake et al. `02, Girvan and Newman `02] | L10. Snapshot Power Law (SPL) [McGlohon et al. `08] |
| Dynamic | L05. Densification Power Law (DPL) [Leskovec et al. `05]<br>L06. Small and shrinking diameter [Albert and Barabási `99, Leskovec et al. `05]<br>L07. Constant size $2^{nd}$ and $3^{rd}$ connected components [McGlohon et al. `08]<br>L08. Principal Eigenvalue Power Law ($\lambda_1$PL) [Akoglu et al. `08]<br>L09. Bursty/self-similar edge/weight additions [Gomez and Santonja `98, Gribble et al. `98, Crovella and Bestavros `99, McGlohon et al. `08] | L11. Weight Power Law (WPL) [McGlohon et al. `08] |

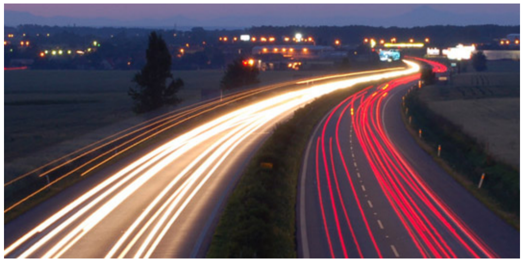
- Mary McGlohon, Leman Akoglu, Christos Faloutsos. *Statistical Properties of Social Networks.* in "Social Network Data Analytics” (Ed.: Charu Aggarwal)




- Deepayan Chakrabarti and Christos Faloutsos, [Graph Mining: Laws, Tools, and Case Studies](#) Oct. 2012, Morgan Claypool.

Carnegie Mellon

# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - …
  - Why so many power-laws?
  - Why no 'good cuts'?
- Part#2: Cascade analysis
- Conclusions

# 2 Questions, one answer

- Q1: why so many power laws
- Q2: why no ‘good cuts’?

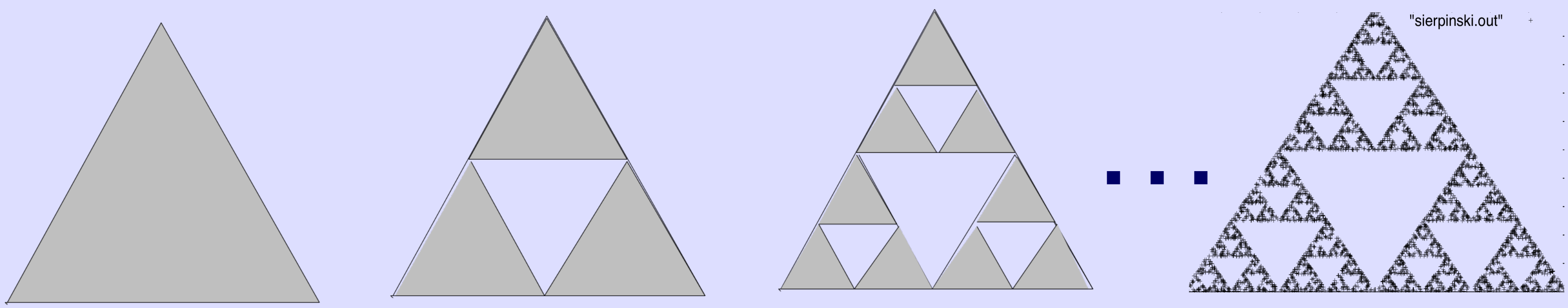
possible

# 2 Questions, one answer

- Q1: why so many power laws
- Q2: why no 'good cuts'?
- A: Self-similarity = fractals = 'RMAT' ~ 'Kronecker graphs'

# 20'' intro to fractals

- Remove the middle triangle; repeat
- -> Sierpinski triangle
- (Bonus question - dimensionality?
  - >1 (inf. perimeter – $(4/3)^{\infty}$ )
  - <2 (zero area – $(3/4)^{\infty}$ )



# 20'' intro to fractals

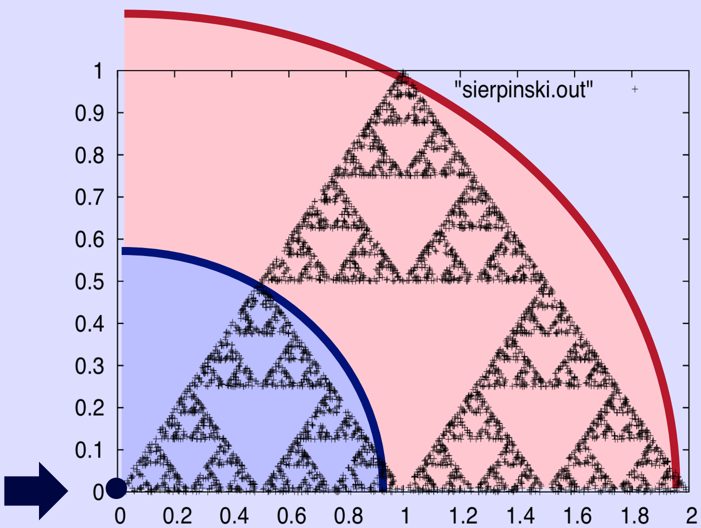
Self-similarity -> no char. scale

-> power laws, eg:

2x the radius,

3x the #neighbors nn(r)

$$nn(r) = C\ r^{\log 3/\log 2}$$



# 20’’ intro to fractals

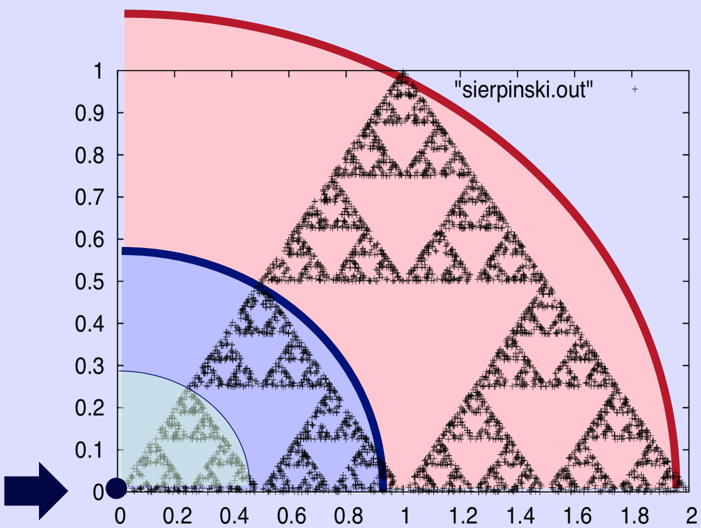
Self-similarity -> <u>no char. scale</u>

-> power laws, eg:

2x the radius,

3x the #neighbors nn(r)

$$nn(r) = C\ r^{\log 3/\log 2}$$

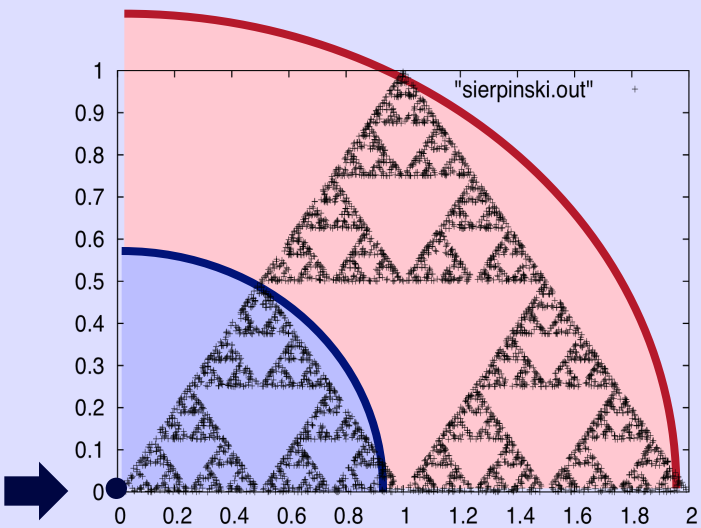


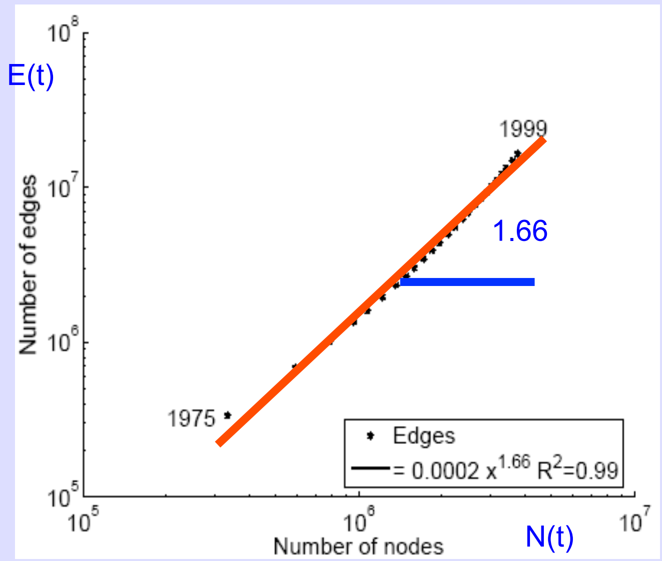
# 20'' intro to fractals

Self-similarity -> no char. scale
-> power laws, eg:
2x the radius,
3x the #neighbors

$nn = C\ r^{\log 3/\log 2}$

Reminder:
Densification P.L.
(2x nodes, ~3x edges)



# 20'' intro to fractals

Self-similarity -> no char. scale

-> power laws, eg:

2x the radius,

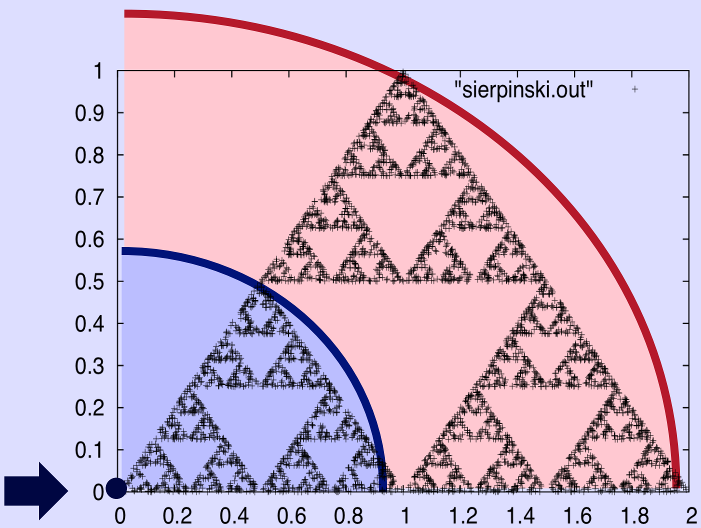
3x the #neighbors

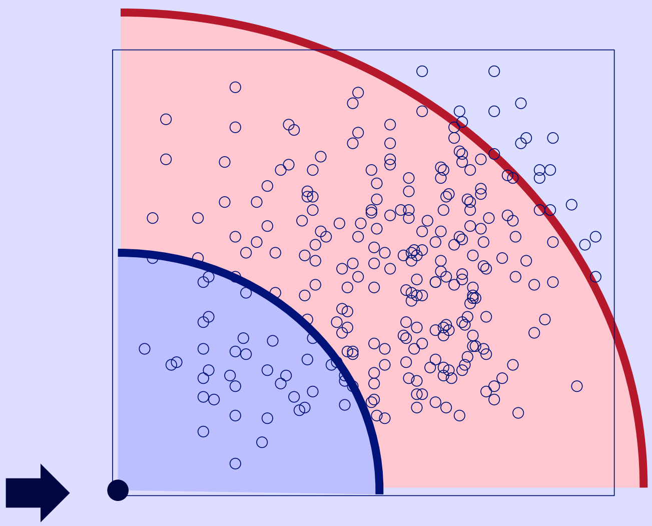
$nn = C\ r^{\log 3/\log 2}$

2x the radius,

4x neighbors

$nn = C\ r^{\log 4/\log 2} = C\ r^{2}$



# 20'' intro to fractals

Self-similarity -> no char. scale

-> power laws, eg:

2x the radius,

3x the #neighbors

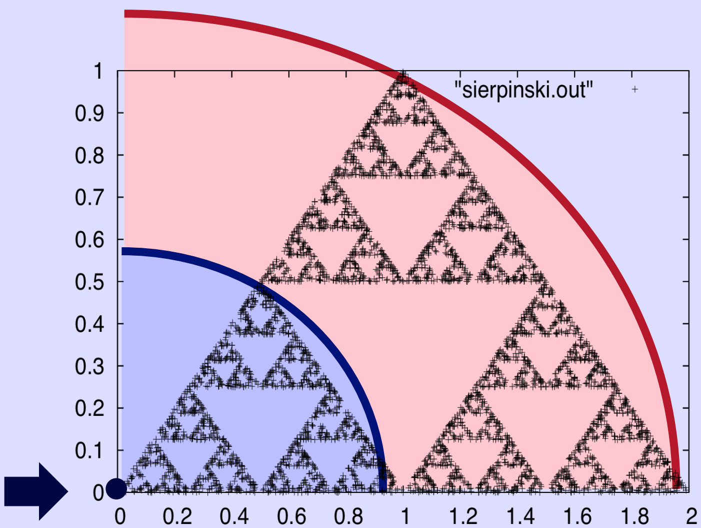
$nn = C\, r^{\log 3/\log 2}$ =1.58

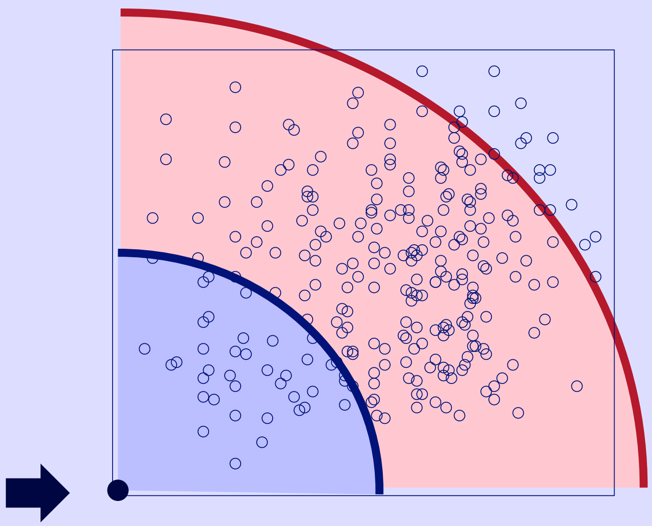
2x the radius,

4x neighbors

$nn = C\, r^{\log 4/\log 2} = C\, r^{2}$

Fractal dim.



# 20'' intro to fractals

**Self-similarity** -> no char. scale

**-> power laws**, eg:

2x the radius,

3x the #neighbors

nn = C r $^{log3/log2}$

2x the radius,

4x neighbors

nn = C r $^{log4/log2}$ = C r $^{2}$

Fractal dim.

# How does self-similarity help in graphs?

- A: RMAT/Kronecker generators
  - With self-similarity, we get all power-laws, automatically,
  - And small/shrinking diameter
  - And `no good cuts'

*R-MAT: A Recursive Model for Graph Mining*,
by D. Chakrabarti, Y. Zhan and C. Faloutsos,
SDM 2004, Orlando, Florida, USA

*Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication,*
by J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, in PKDD 2005, Porto, Portugal

Carnegie Mellon

# Graph gen.: Problem dfn

- Given a growing graph with count of nodes $N_1$, $N_2$, …
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns

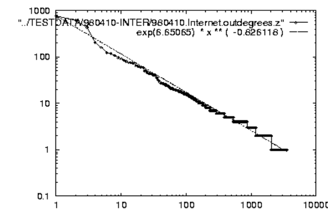
    S1 Power Law Degree Distribution

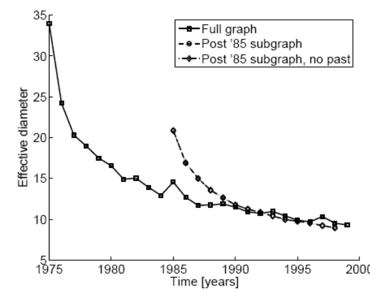
    S2 Power Law eigenvalue and eigenvector distribution

    Small Diameter
  - Dynamic Patterns

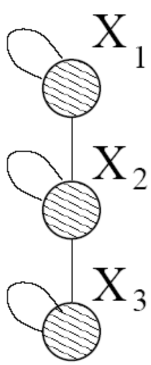
    T2 Growth Power Law (2x nodes; 3x edges)

    T1 Shrinking/Stabilizing Diameters



# Kronecker Graphs



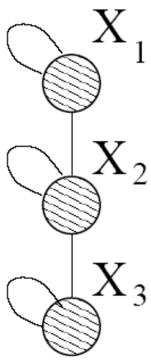

| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

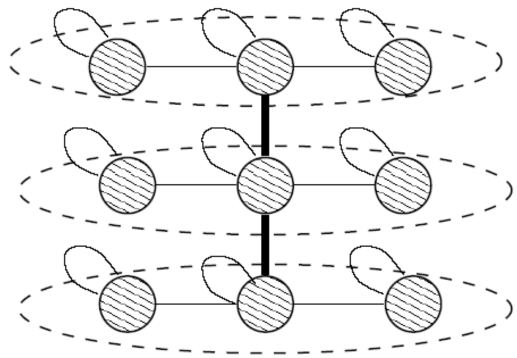
$G_1$

Adjacency matrix

Carnegie Mellon

# Kronecker Graphs



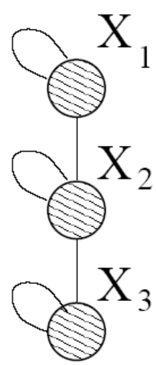
Intermediate stage

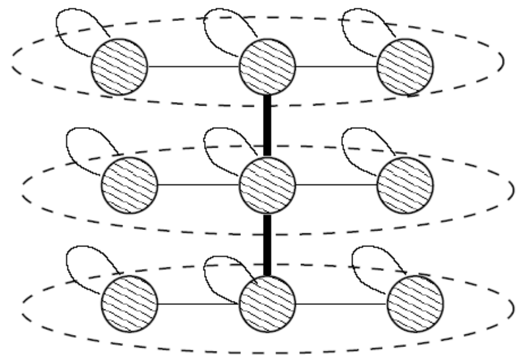
| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

$G_1$

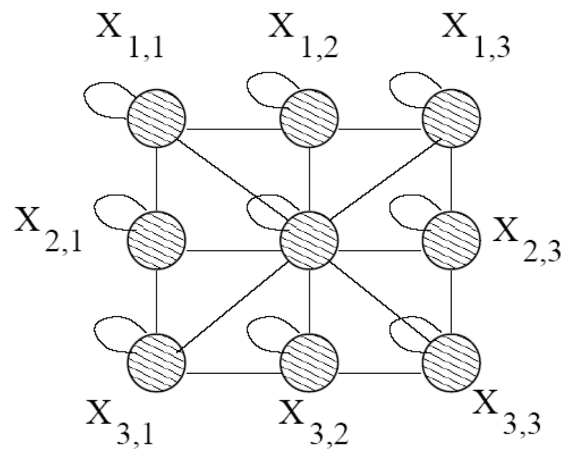
Adjacency matrix

# Kronecker Graphs



Intermediate stage




| | | |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |

$G_1$

Adjacency matrix

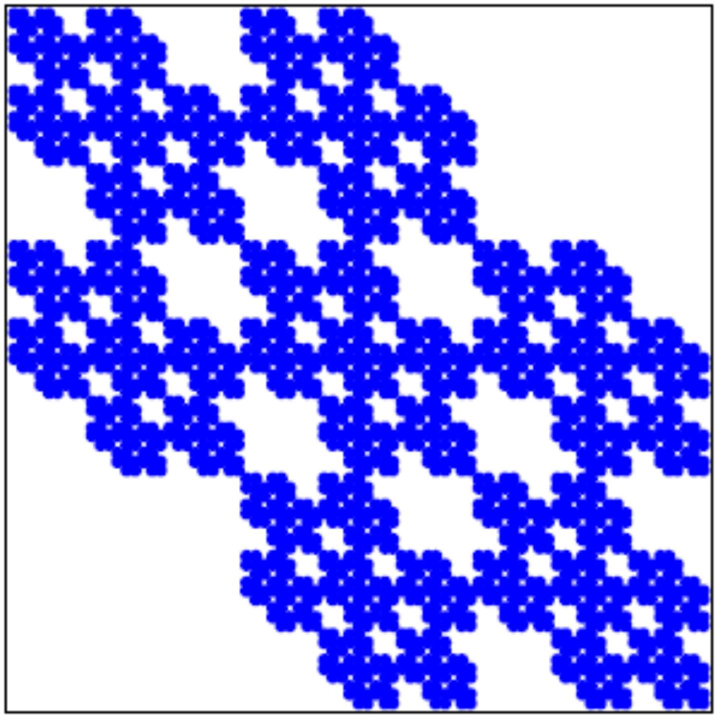
| | | |
|---|---|---|
| $G_1$ | $G_1$ | 0 |
| $G_1$ | $G_1$ | $G_1$ |
| 0 | $G_1$ | $G_1$ |

$G_2 = G_1 \otimes G_1$

Adjacency matrix

# Kronecker Graphs

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

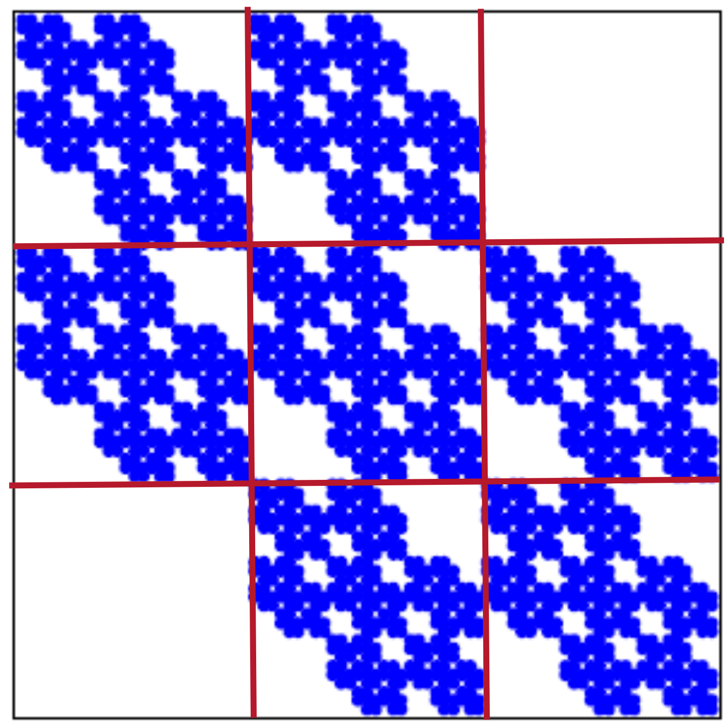


$G_4$ adjacency matrix

Carnegie Mellon

# Kronecker Graphs

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

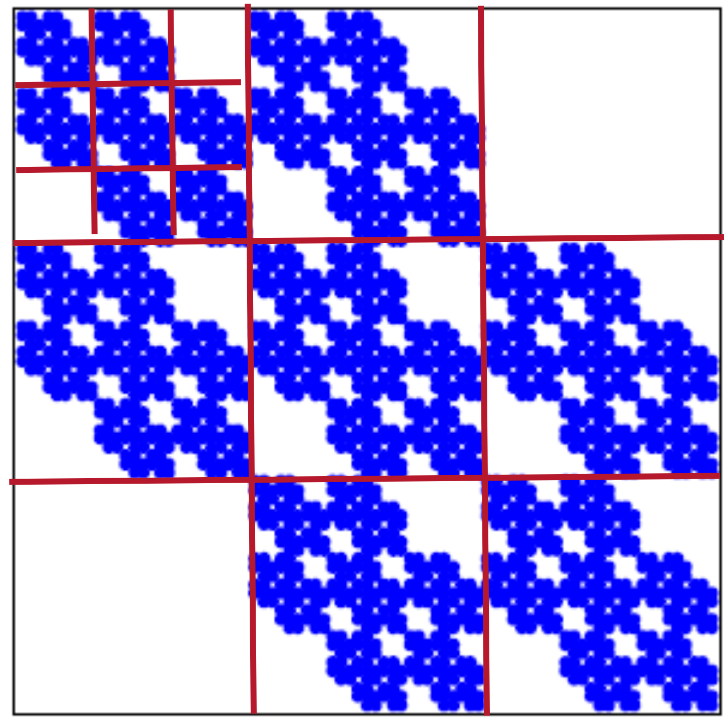


$G_4$ adjacency matrix

Carnegie Mellon

# Kronecker Graphs

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



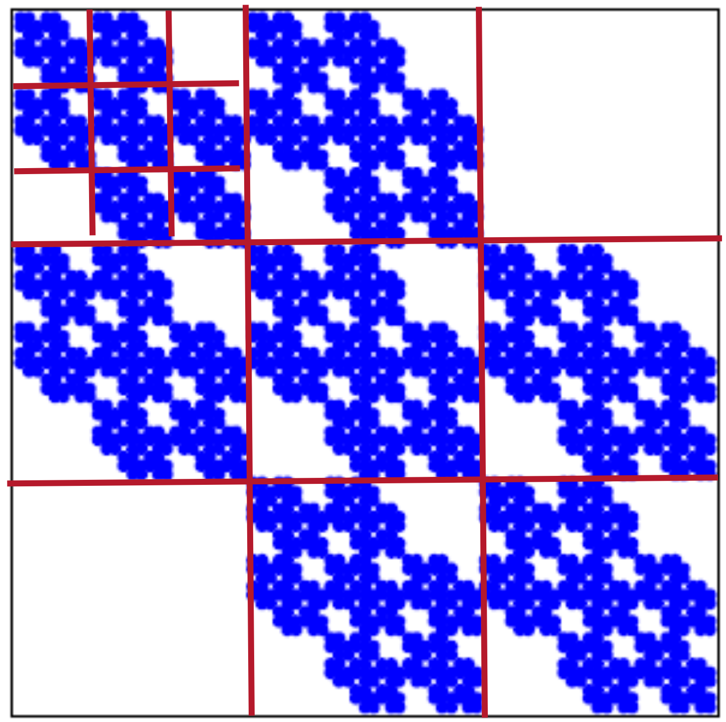
$G_4$ adjacency matrix

Carnegie Mellon

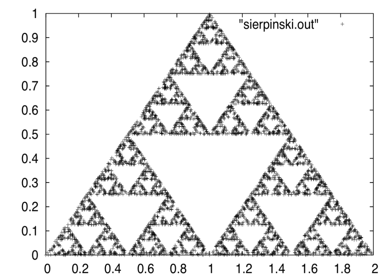
# Kronecker Graphs

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

Holes within holes; Communities within communities



$G_4$ adjacency matrix



# Properties:

- We can PROVE that
  - Degree distribution is multinomial ~ power law
  - new – Diameter: constant
  - Eigenvalue distribution: multinomial
  - First eigenvector: multinomial

# Problem Definition

- Given a growing graph with nodes $N_1$, $N_2$, …
- Generate a realistic sequence of graphs that will obey all the patterns
  - Static Patterns
    - ✓ Power Law Degree Distribution
    - ✓ Power Law eigenvalue and eigenvector distribution
    - ✓ Small Diameter
  - Dynamic Patterns
    - ✓ Growth Power Law
    - ✓ Shrinking/Stabilizing Diameters
- First generator for which we can **prove** all these properties

Carnegie Mellon

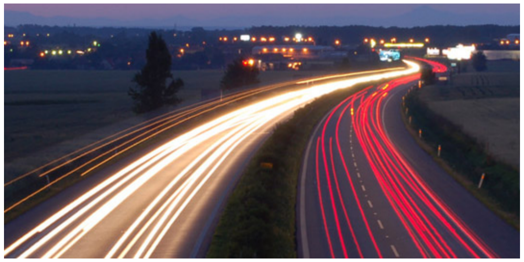
# Impact: Graph500

- Based on RMAT (= 2x2 Kronecker)
- Standard for graph benchmarks
- http://www.graph500.org/
- Competitions 2x year, with all major entities: LLNL, Argonne, ITC-U. Tokyo, Riken, ORNL, Sandia, PSC, …

*To iterate is human, to recurse is devine*

*R-MAT: A Recursive Model for Graph Mining*, by D. Chakrabarti, Y. Zhan and C. Faloutsos, SDM 2004, Orlando, Florida, USA

# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - …
  - Q1: Why so many power-laws?
  - Q2: Why no ‘good cuts’?
- Part#2: Cascade analysis
- Conclusions

A: real graphs -> self similar -> power laws

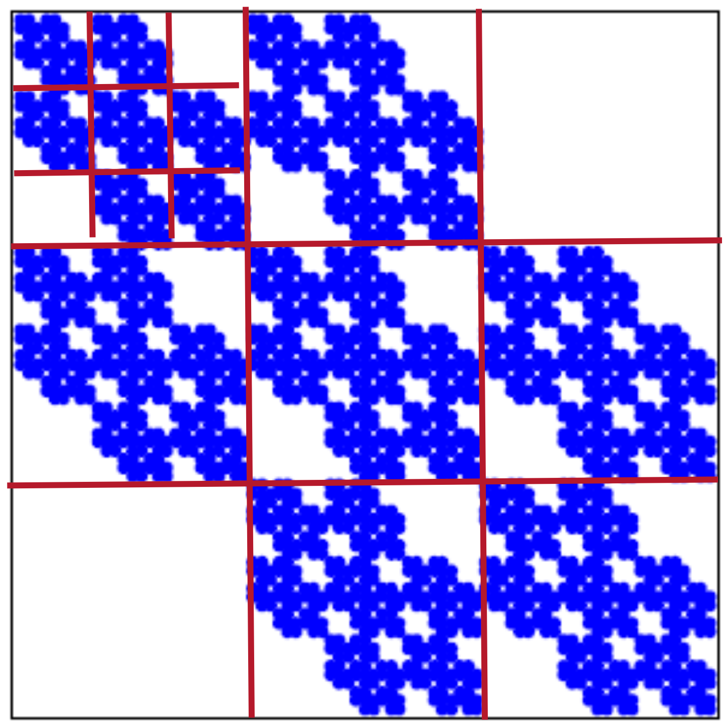
# Q2: Why ‘no good cuts’?

- A: self-similarity
  - Communities within communities within communities …

REMINDER

# Kronecker Product – a Graph

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



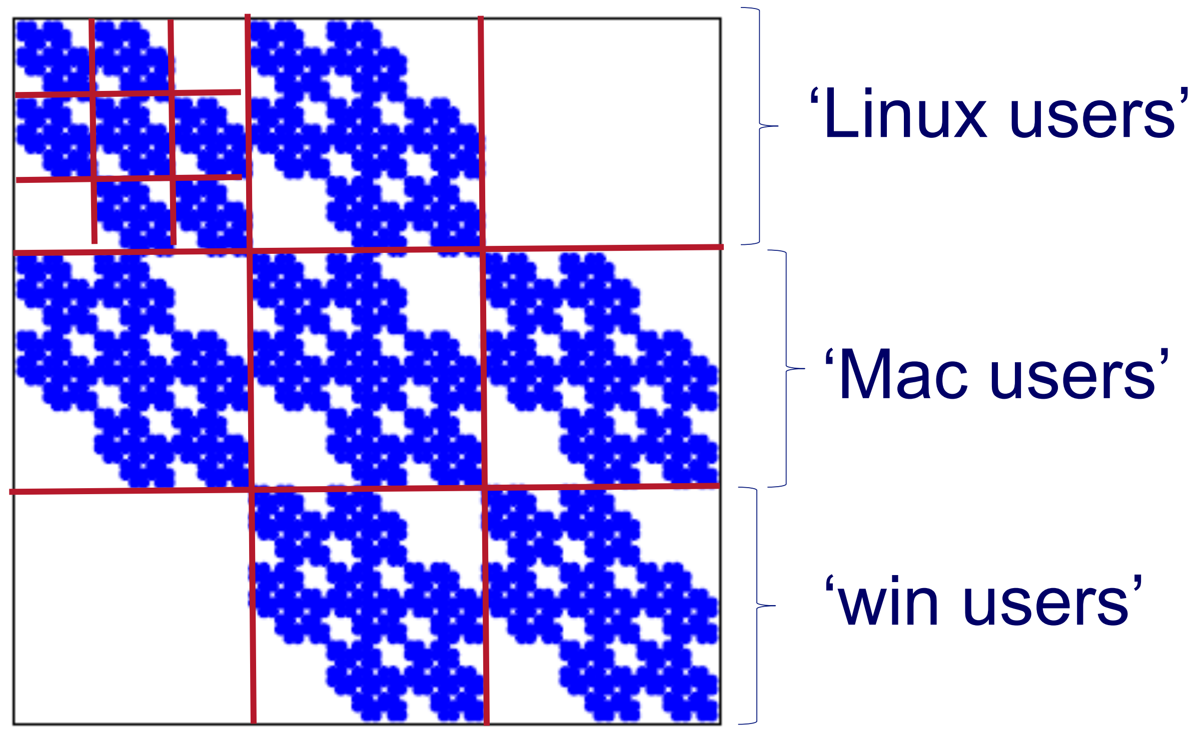
$G_4$ adjacency matrix

REMINDER

# Kronecker Product – a Graph

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

Communities within communities within communities …



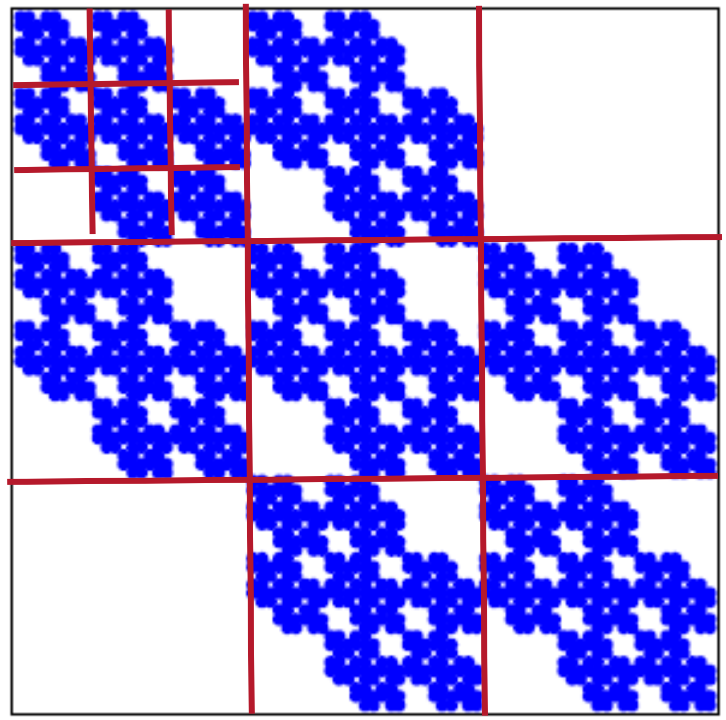

$G_4$ adjacency matrix

REMINDER

# Kronecker Product – a Graph

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

Communities within communities within communities …



How many Communities?
3?
9?
27?

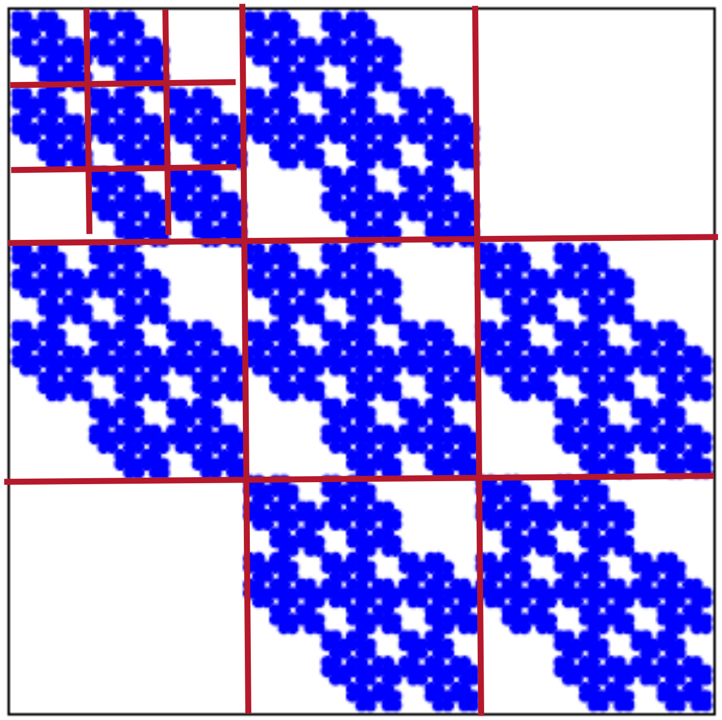
$G_4$ adjacency matrix

REMINDER

# Kronecker Product – a Graph

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

Communities within communities within communities …



$G_4$ adjacency matrix

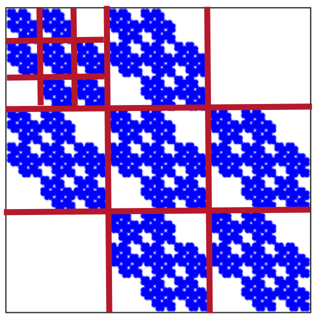
How many Communities?
3?
9?
27?

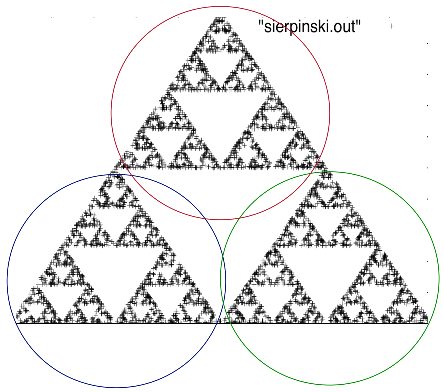
A: one – but not a typical, block-like community…

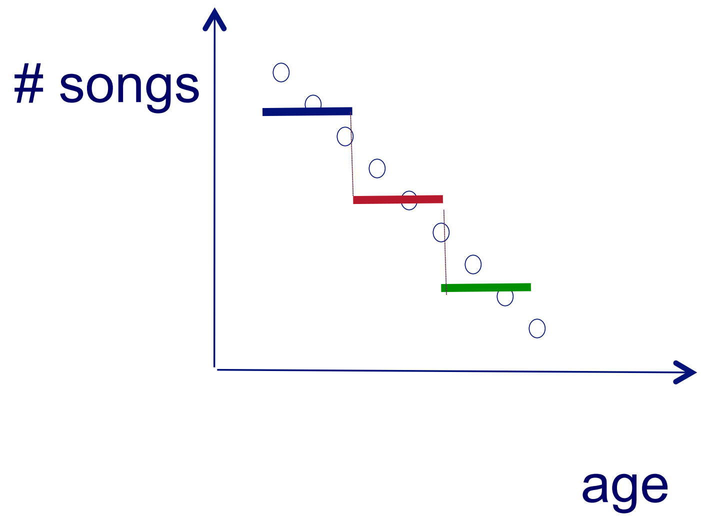
Communities?

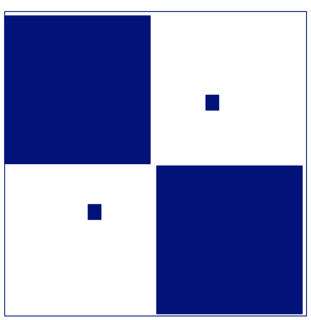
(Gaussian) Clusters?

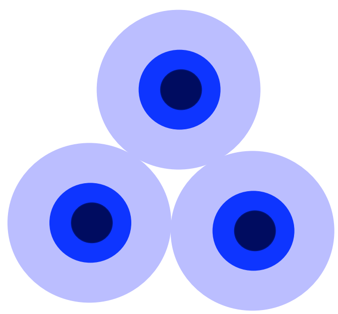
Piece-wise flat parts?



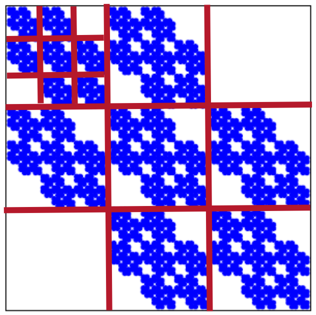


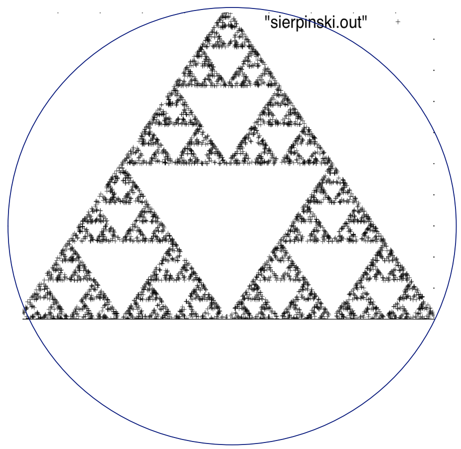



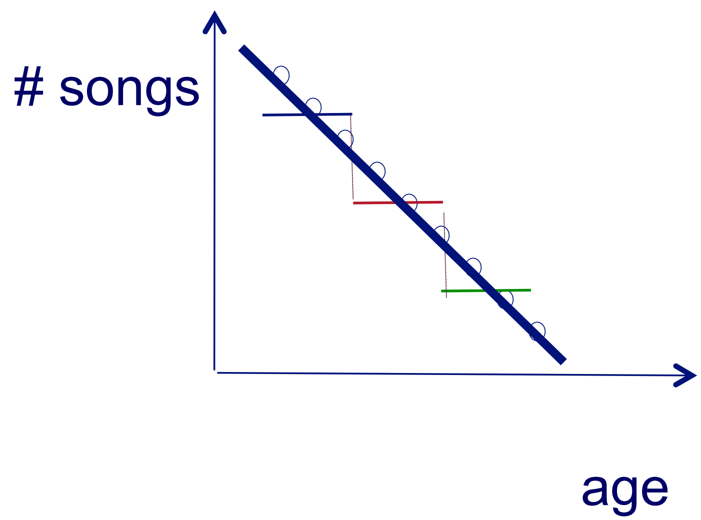



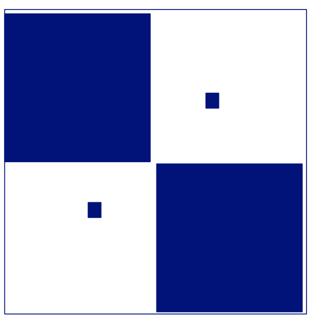


Carnegie Mellon





Wrong questions to ask!

Carnegie Mellon

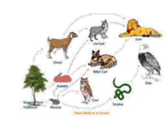
# Summary of Part#1

- *many* patterns in real graphs
  - Small & shrinking diameters
  - Power-laws everywhere
  - Gaussian trap
  - ‘no good cuts’
- Self-similarity (RMAT/Kronecker): good model

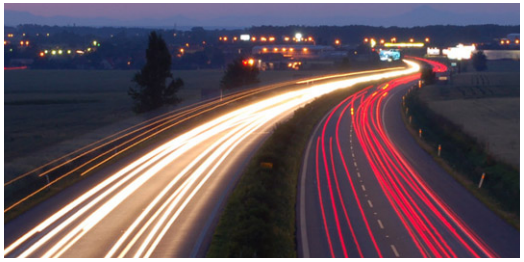
# Part 2: Cascades & Immunization

# Why do we care?

- Information Diffusion
- Viral Marketing
- Epidemiology and Public Health
- Cyber Security
- Human mobility
- Games and Virtual Worlds
- Ecology
- ........



# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
  - (Fractional) Immunization
  - Epidemic thresholds
- Conclusions

# *Fractional Immunization of Networks*

B. Aditya Prakash, Lada Adamic, Theodore Iwashyna (M.D.), Hanghang Tong,
Christos Faloutsos

SDM 2013, Austin, TX

# Whom to immunize?

- Dynamical Processes over networks

- Each circle is a hospital
- ~3,000 hospitals
- More than 30,000 patients transferred

[US-MEDICARE NETWORK 2005]

**Problem**: Given *k* units of disinfectant, whom to immunize?

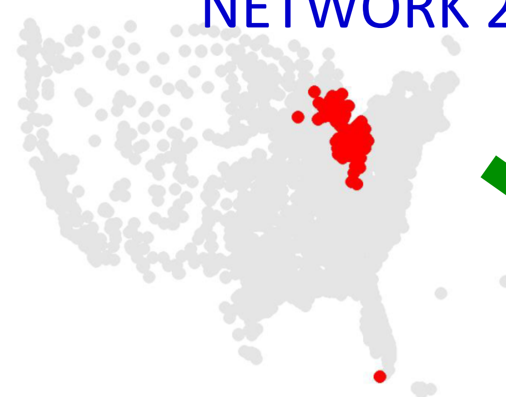
Carnegie Mellon

# Whom to immunize?

**~6x fewer!**

CURRENT PRACTICE

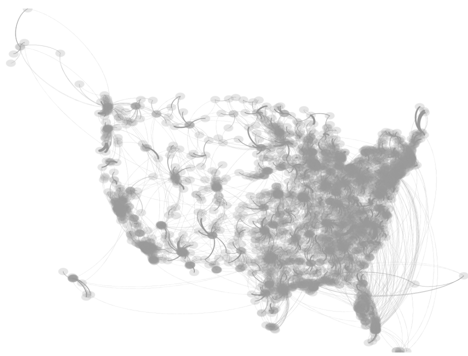
[US-MEDICARE NETWORK 2005]



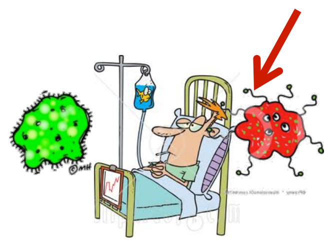
OUR METHOD

Hospital-acquired inf. : 99K+ lives, $5B+ per year

# Fractional Asymmetric Immunization



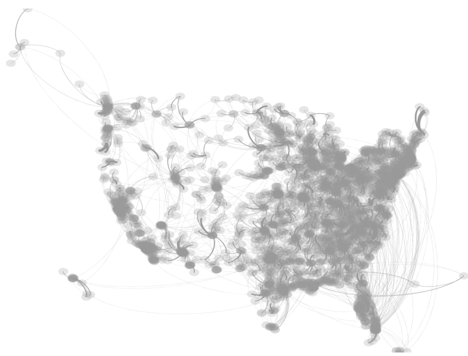
Drug-resistant Bacteria
(like XDR-TB)

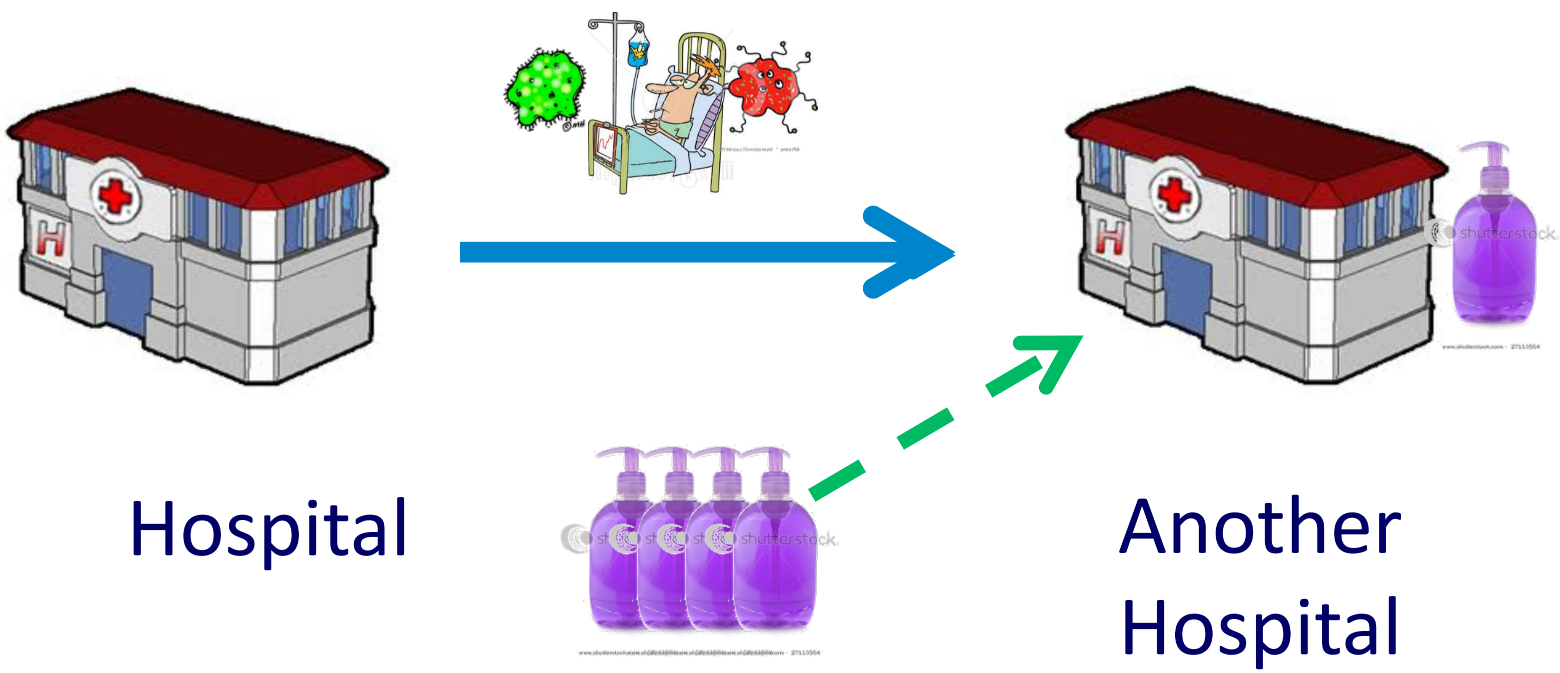


Hospital

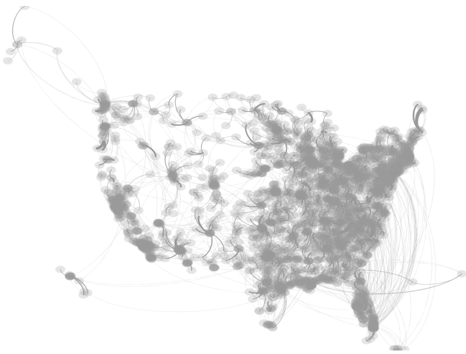
Another Hospital

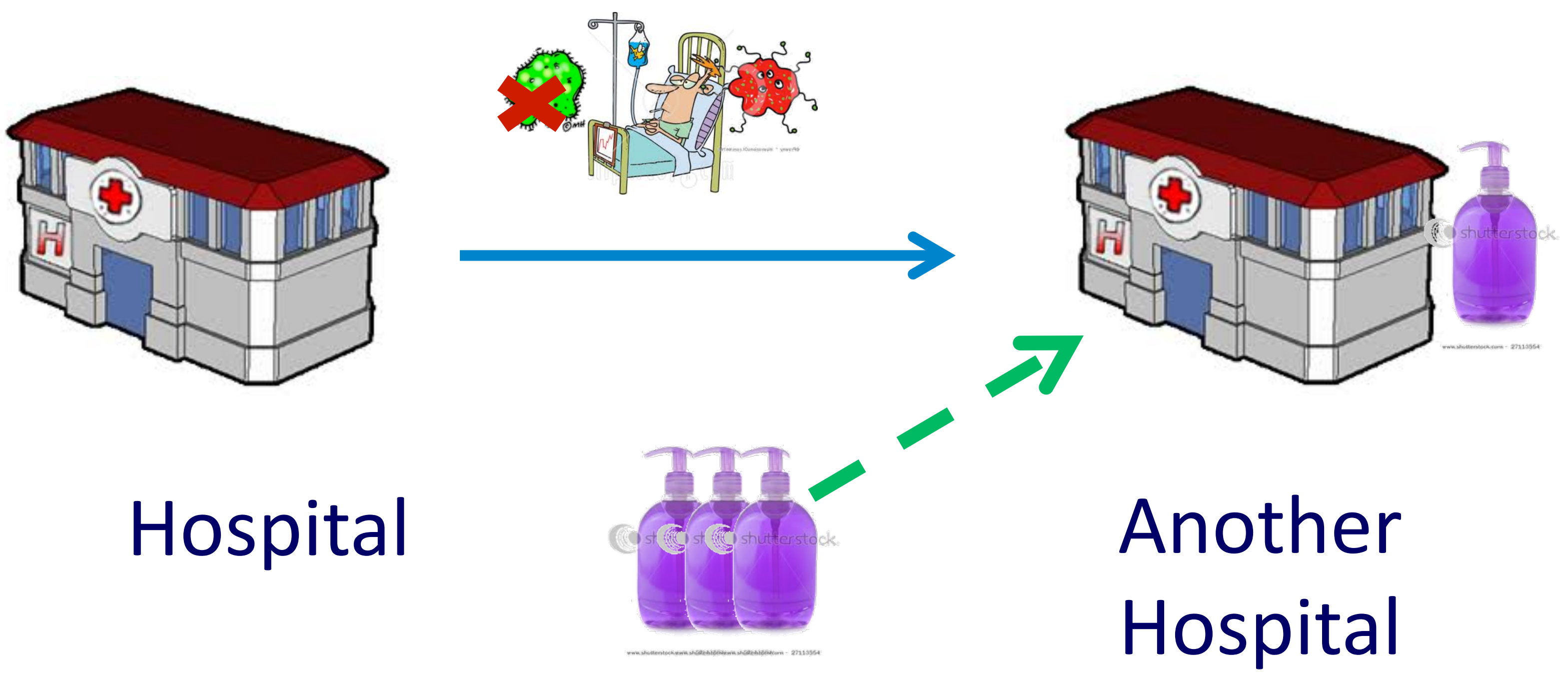
# Fractional Asymmetric Immunization



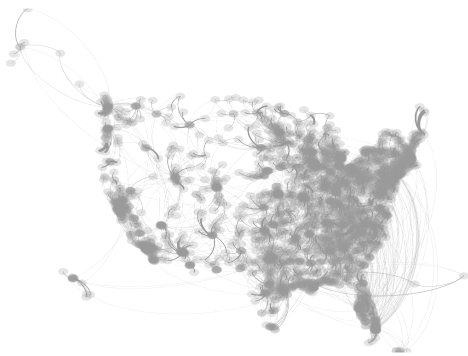


# Fractional Asymmetric Immunization

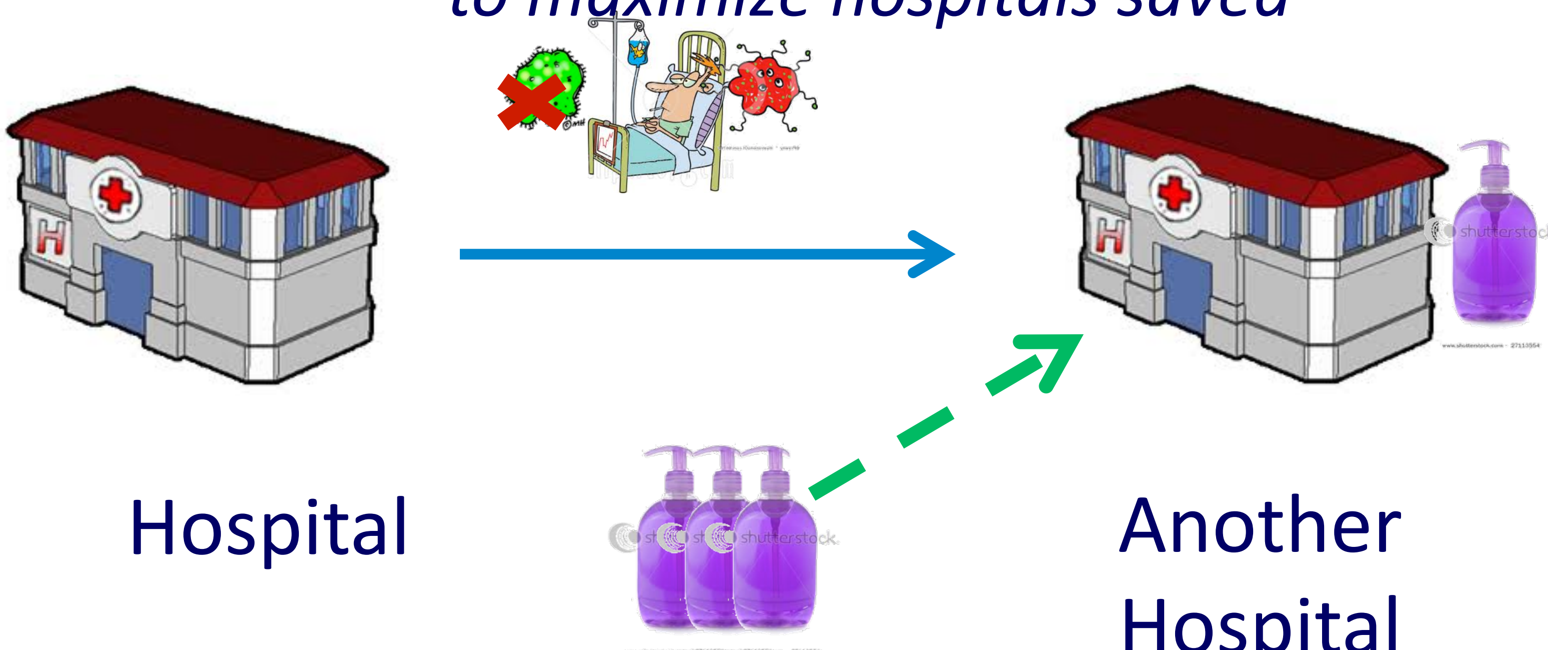




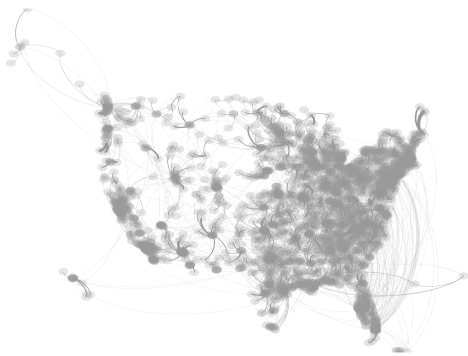
# Fractional Asymmetric Immunization



***Problem**:*
*Given k units of disinfectant, distribute them to maximize hospitals saved*

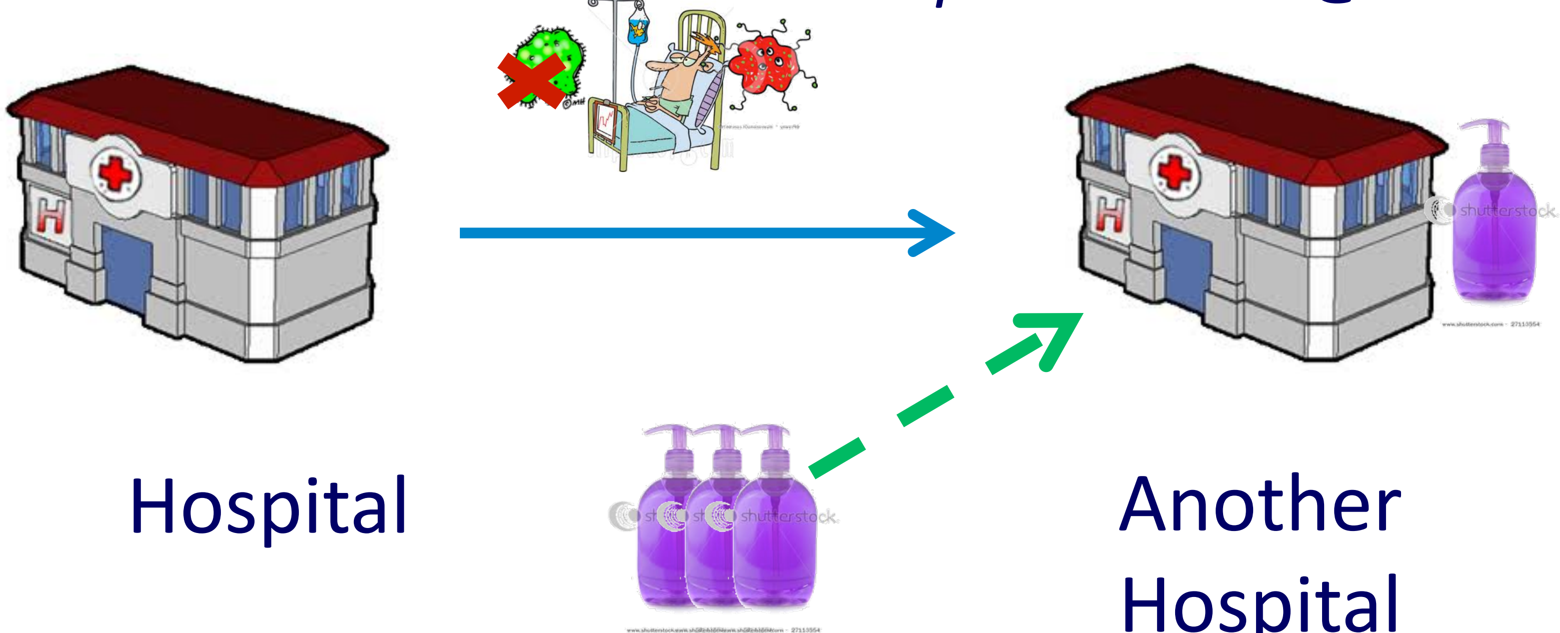


# Fractional Asymmetric Immunization



***Problem**:*

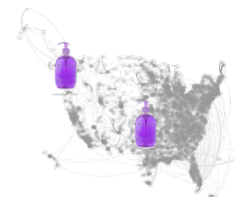
*Given k units of disinfectant,*
*distribute them*
*to maximize hospitals saved @ 365 days*



# Straightforward solution:

Simulation:

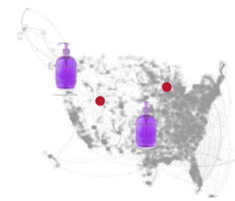
1. Distribute resources
2. 'infect' a few nodes
3. Simulate evolution of spreading
   – (10x, take avg)
4. Tweak, and repeat step 1



# Straightforward solution:

Simulation:

1. Distribute resources
2. ‘infect’ a few nodes
3. Simulate evolution of spreading
   – (10x, take avg)
4. Tweak, and repeat step 1

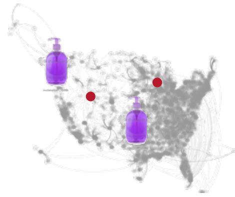


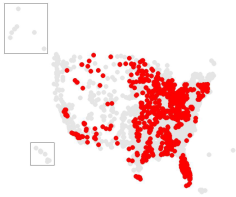
Carnegie Mellon

# Straightforward solution:

Simulation:

1. Distribute resources
2. 'infect' a few nodes
3. Simulate evolution of spreading
   - (10x, take avg)
4. Tweak, and repeat step 1

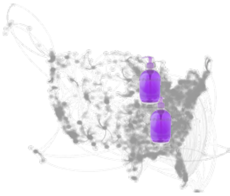


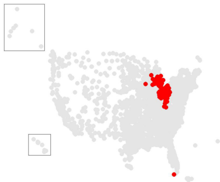


# Straightforward solution:

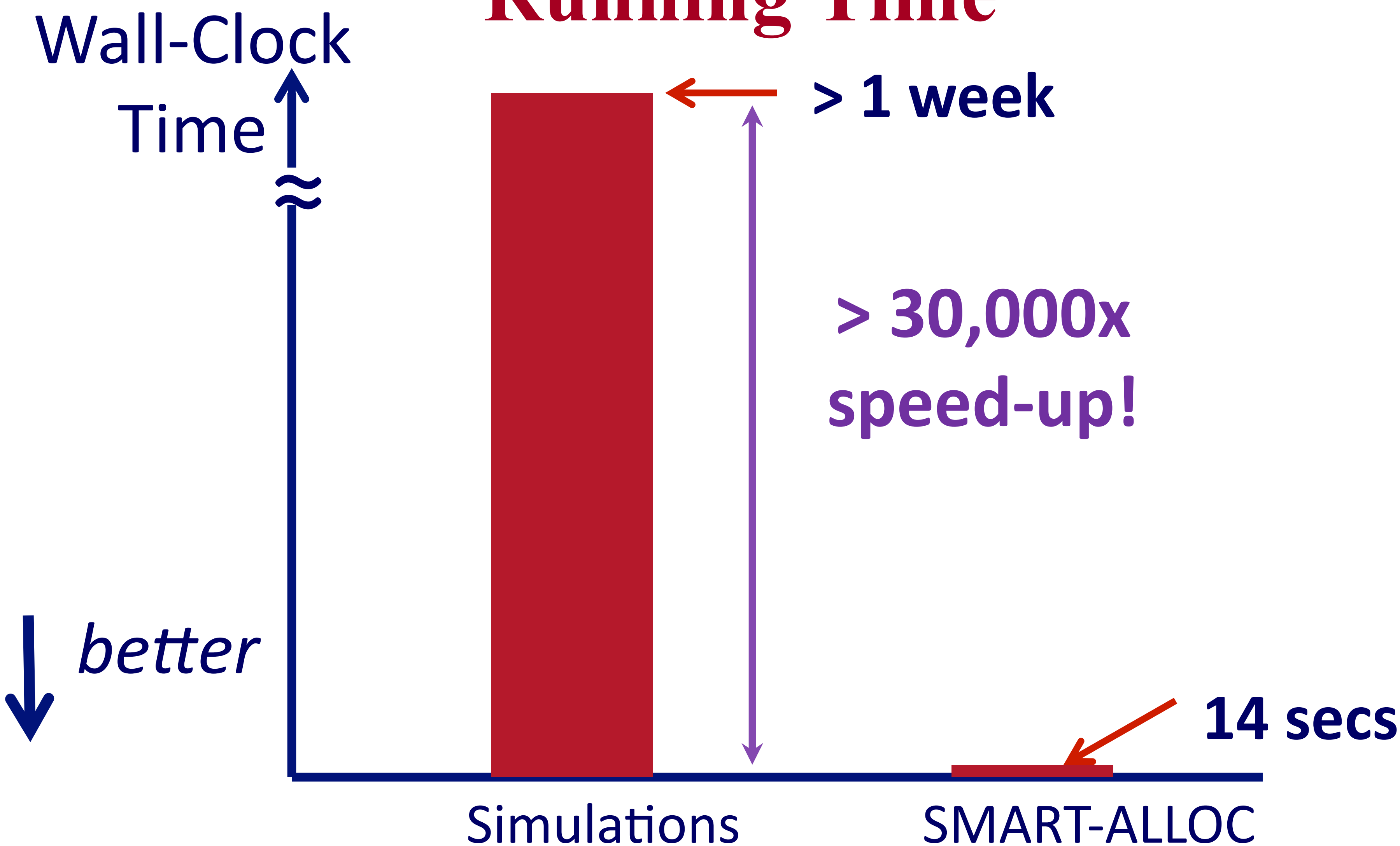
Simulation:

1. Distribute resources
2. 'infect' a few nodes
3. Simulate evolution of spreading
   – (10x, take avg)
4. Tweak, and repeat step 1



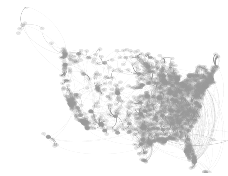


# Running Time

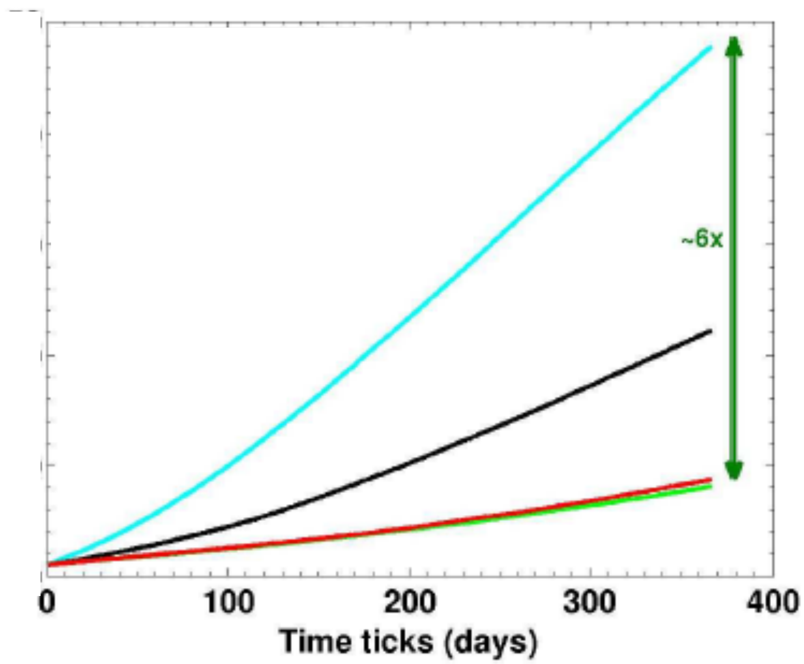


Carnegie Mellon

# Experiments



# infected




uniform

*better*

SMART-ALLOC

# epochs

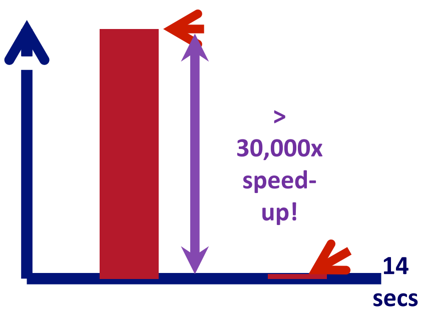
K = 120

# What is the ‘silver bullet’?

A: Try to decrease connectivity of graph

Q: how to measure connectivity?

- Avg degree? Max degree?
- Std degree / avg degree ?
- Diameter?
- Modularity?
- ‘Conductance’ (~min cut size)?
- Some combination of above?



# What is the 'silver bullet'?

A: Try to decrease connectivity of graph

Q: how to measure connectivity?

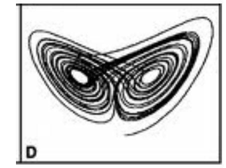
A: first **eigenvalue** of adjacency matrix

Avg degree
Max degree
Diameter
Modularity
'Conductance'

Q1: why??

(Q2: dfn & intuition of eigenvalue ? )

Carnegie Mellon

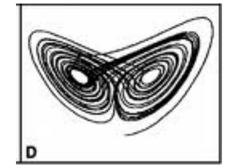


# Why eigenvalue?

A1: ‘G2’ theorem and ‘**eigen-drop**’:

- For (almost) **any** type of virus
- For **any** network
- -> no epidemic, if small-enough first eigenvalue ($\lambda_1$) of *adjacency* matrix

*Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks*, B. Aditya Prakash, Deepayan Chakrabarti, Michalis Faloutsos, Nicholas Valler, Christos Faloutsos, ICDM 2011, Vancouver, Canada




# Why eigenvalue?

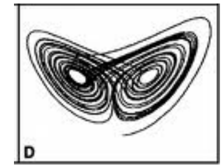
A1: ‘G2’ theorem and ‘**eigen-drop**’:

- For (almost) **any** type of virus
- For **any** network
- -> no epidemic, if small-enough first eigenvalue ($\lambda_1$) of *adjacency* matrix

- Heuristic: for immunization, try to min $\lambda_1$
- The smaller $\lambda_1$, the closer to extinction.

Carnegie Mellon

# G2 theorem




*Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks*

B. Aditya Prakash, Deepayan Chakrabarti, Michalis Faloutsos, Nicholas Valler, Christos Faloutsos

IEEE ICDM 2011, Vancouver

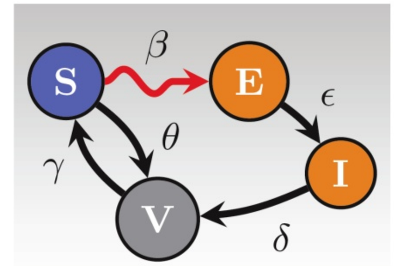
extended version, in arxiv

http://arxiv.org/abs/1004.0060

~10 pages proof

# Our thresholds for some models

- $s$ = *effective strength*
- $s < 1$ : *below threshold*

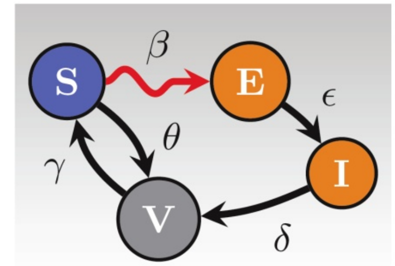



| Models | Effective Strength (s) | Threshold (tipping point) |
| --- | --- | --- |
| SIS, SIR, SIRS, SEIR | $s = \lambda \cdot \left(\frac{\beta}{\delta}\right)$ | $s = 1$ |
| SIV, SEIV | $s = \lambda \cdot \left(\frac{\beta\gamma}{\delta(\gamma+\theta)}\right)$ | |
| $SI_1I_2V_1V_2$ (**H.I.V.**) | $s = \lambda \cdot \left(\frac{\beta_1 v_2 + \beta_2 \varepsilon}{v_2(\varepsilon + v_1)}\right)$ | |

Carnegie Mellon

# Our thresholds for some models

- *s = effective strength*
- *s < 1 : below threshold*



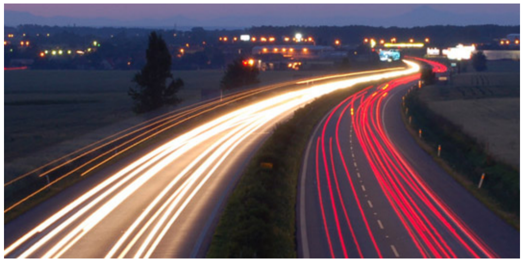

No immunity

Temp. immunity

w/ incubation

| | e Strength | Threshold (tipping point) |
|---|---|---|
| SIS, SIR, SIRS, SEIR | $s = \lambda \cdot \left(\frac{\beta}{\delta}\right)$ | $s = 1$ |
| SIV, SEIV | $s = \lambda \cdot \left(\frac{\beta\gamma}{\delta(\gamma+\theta)}\right)$ | |
| $SI_1I_2V_1V_2$ (**H.I.V.**) | $s = \lambda \cdot \left(\frac{\beta_1 v_2 + \beta_2 \varepsilon}{v_2(\varepsilon + v_1)}\right)$ | |

# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
  - (Fractional) Immunization
  - intuition behind $\lambda_1$
- Conclusions

# Intuition for λ

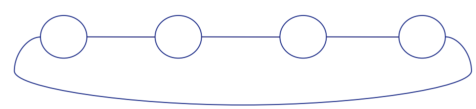
## "Official" definitions:

- *Let **A** be the adjacency matrix. Then λ is the root with the largest magnitude of the characteristic polynomial of **A** [det(**A** – x**I**)].*
- Also: $\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$

Neither gives much intuition!

## "Un-official" Intuition

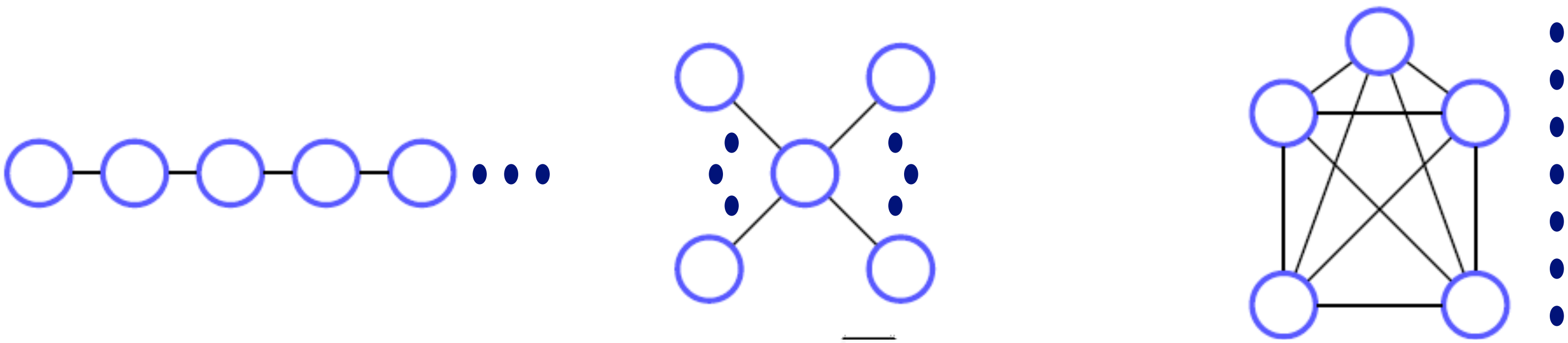
- For 'homogeneous' graphs, λ == *degree*



- λ ~ avg degree
  - done right, for skewed degree distributions

# Largest Eigenvalue (λ)

better connectivity ⟶ higher $\lambda$



$\lambda \approx 2$ $\lambda = \sqrt{N}$ $\lambda = N-1$

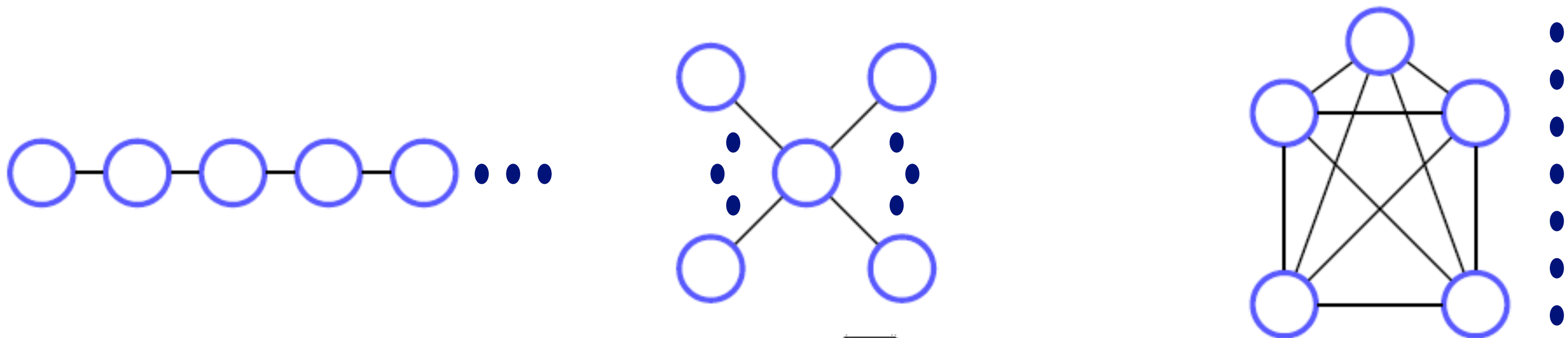
(a)Chain (b)Star (c)Clique

$\lambda \approx 2$ $\lambda = 31.67$ $\lambda = 999$

$N$ = 1000 nodes

# Largest Eigenvalue (λ)

better connectivity ⟶ higher $\lambda$



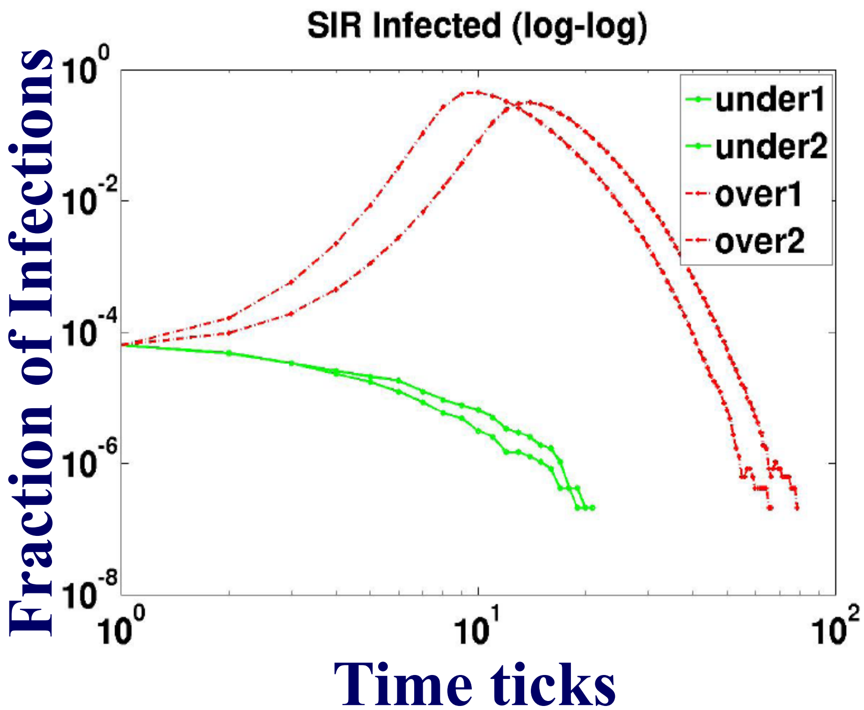
$\lambda \approx 2$ $\lambda = \sqrt{N}$ $\lambda = N\text{-}1$

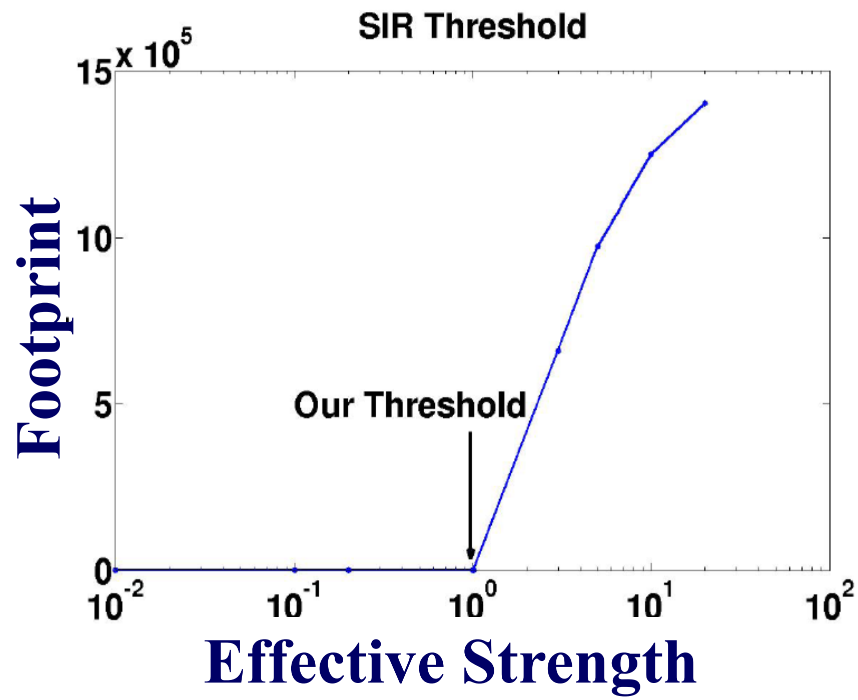
(a)Chain (b)Star (c)Clique

$\lambda \approx 2$ $\lambda = 31.67$ $\lambda = 999$

$N$ = 1000 nodes

# Examples: Simulations – SIR (mumps)



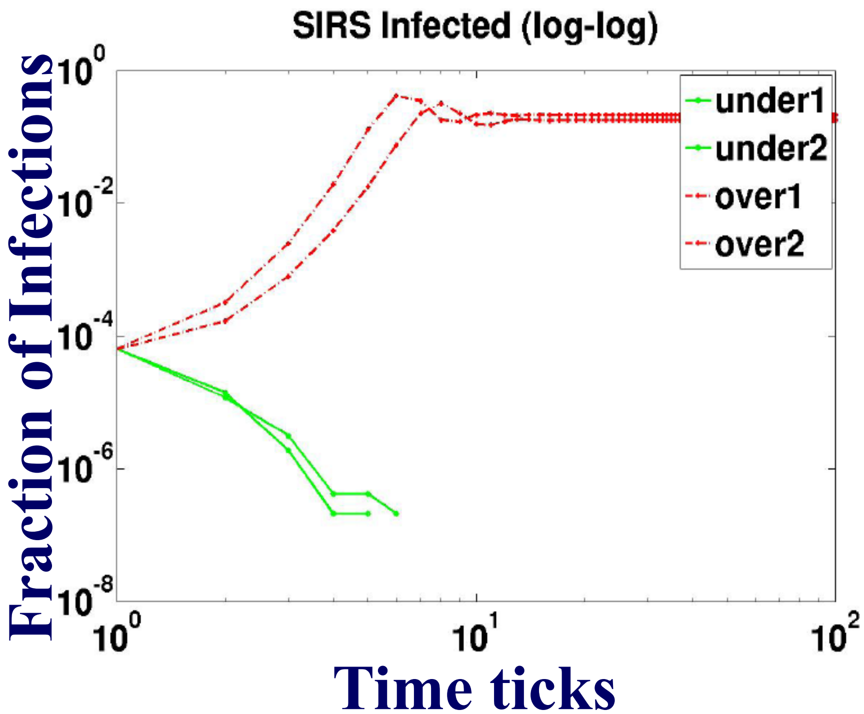

**(a) Infection profile** **(b) "Take-off" plot**

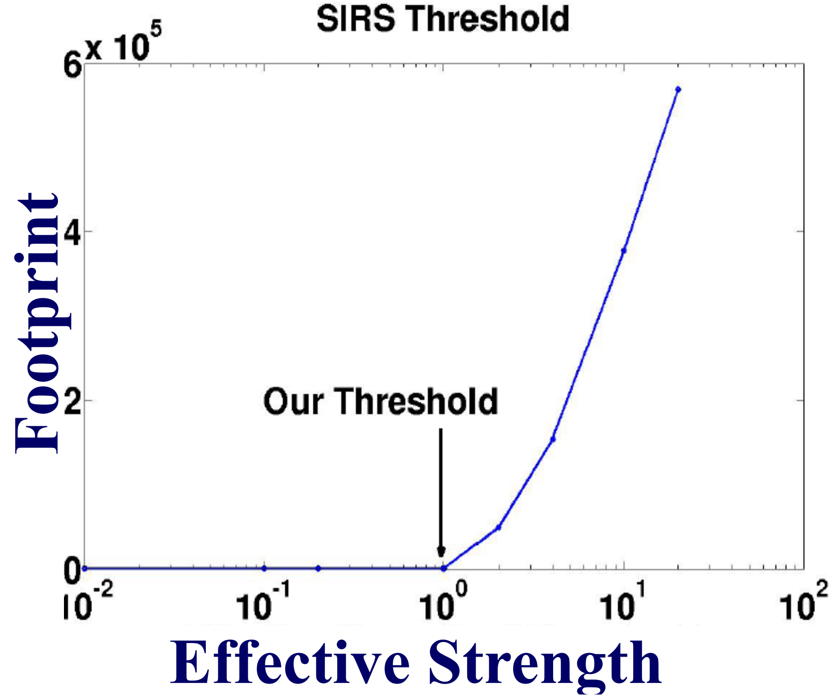
PORTLAND graph: *synthetic population,*

*31 million links, 6 million nodes*

Carnegie Mellon

# Examples: Simulations – SIRS (pertusis)




**(a) Infection profile** **(b) "Take-off" plot**

PORTLAND graph: *synthetic population,*

*31 million links, 6 million nodes*

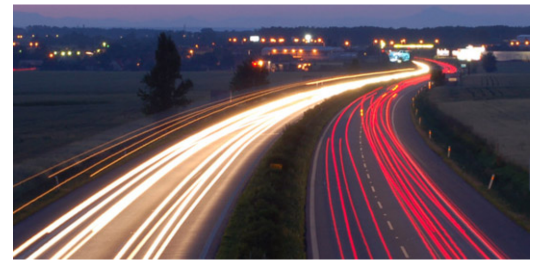
# Immunization - conclusion

In (**almost any**) immunization setting,

- Allocate resources, such that to
- **Minimize** $\lambda_1$
- (*regardless* of virus specifics)

- Conversely, in a market penetration setting
  - Allocate resources to
  - Maximize $\lambda_1$

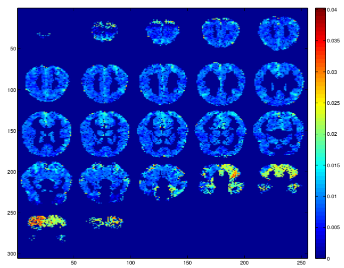
# Roadmap

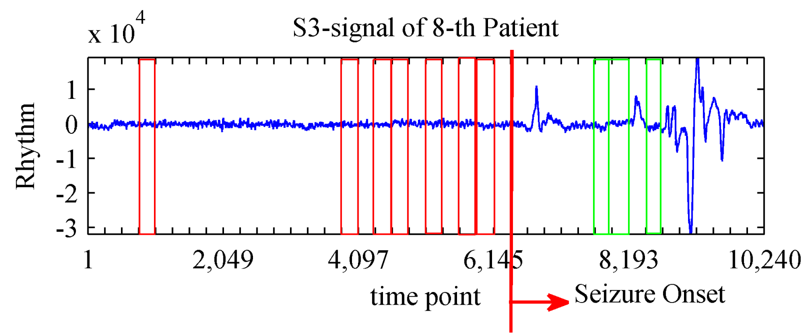


- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
  - (Fractional) Immunization
  - Epidemic thresholds
- ➡ What next?
- Acks & Conclusions
- [Tools: ebay fraud; tensors; spikes]

# Challenge #1: 'Connectome' – brain wiring

- Which neurons get activated by 'bee'
- How wiring evolves
- Modeling epilepsy





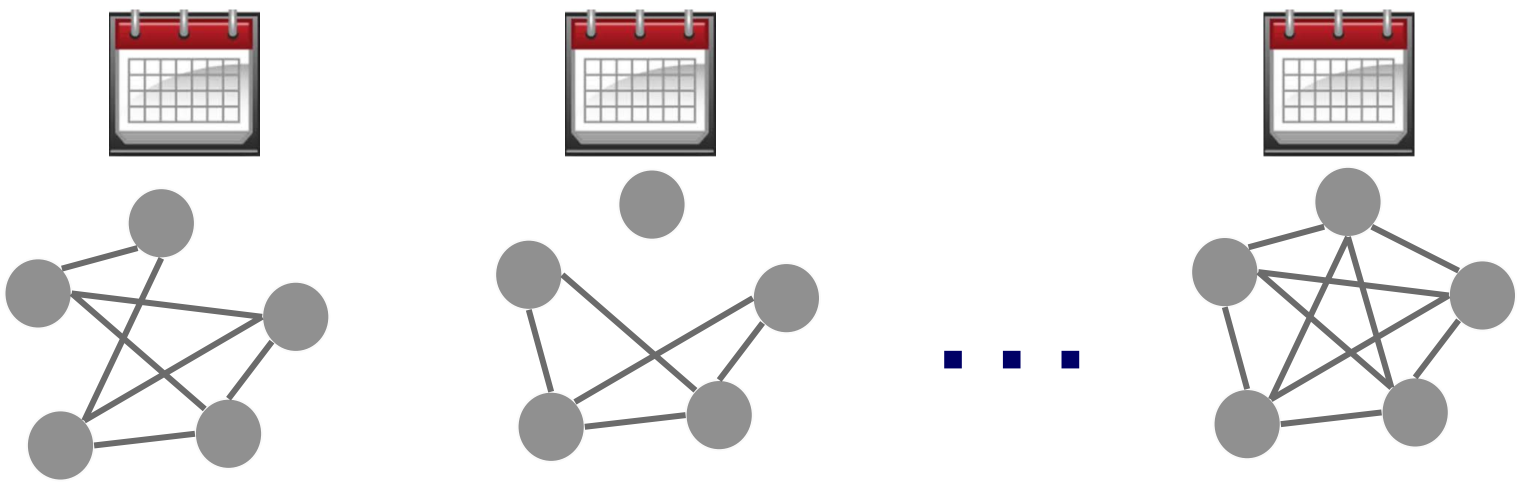

Tom Mitchell George Karypis

N. Sidiropoulos V. Papalexakis

Carnegie Mellon

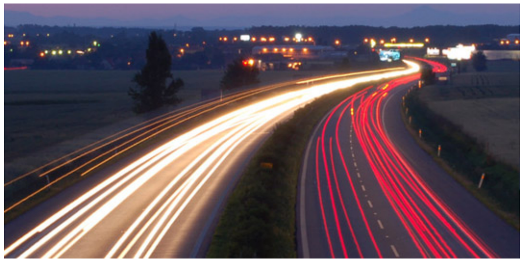
# Challenge#2: Time evolving networks / tensors

- Periodicities? Burstiness?
- What is ‘typical’ behavior of a node, over time
- Heterogeneous graphs (= nodes w/ attributes)



Carnegie Mellon

# Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: Cascade analysis
  - (Fractional) Immunization
  - Epidemic thresholds
- Acks & Conclusions
- [Tools: ebay fraud; tensors; spikes]

Off line

Carnegie Mellon

# Thanks

*Disclaimer: All opinions are mine; not necessarily reflecting the opinions of the funding agencies*

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

# Project info: PEGASUS

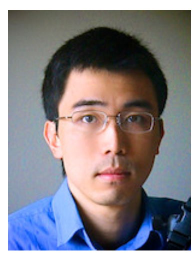
www.cs.cmu.edu/~pegasus

Results on large graphs: with Pegasus + hadoop + M45

Apache license

Code, papers, manual, video

Prof. U Kang



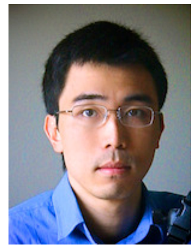
Prof. Polo Chau

Carnegie Mellon

# Cast

Akoglu, Leman

Beutel, Alex



Chau, Polo

Kang, U

Koutra, Danai

Lee, Jay Yoon

Prakash, Aditya

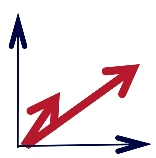
Papalexakis, Vagelis

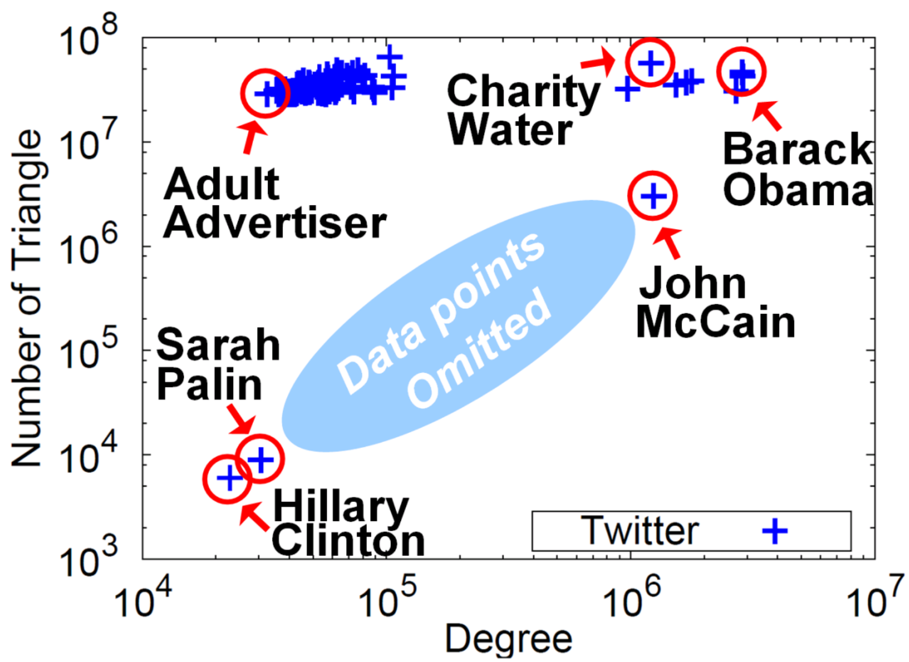
Shah, Neil

Tong, Hanghang

# CONCLUSION#1 – Big data

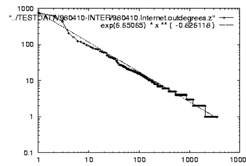
- **Large** datasets reveal patterns/outliers that are invisible otherwise

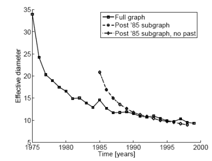



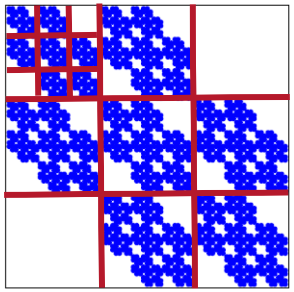


# CONCLUSION#2 – self-similarity

- powerful tool / viewpoint
  - Power laws; shrinking diameters
  - **Gaussian trap** (eg., F.O.F.)
  - ‘no good cuts’
  - RMAT – graph500 generator



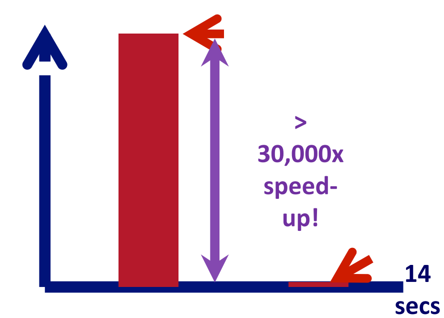
# CONCLUSION#3 – eigen-drop

- Cascades & immunization: G2 theorem & **eigenvalue**


[US-MEDICARE NETWORK 2005]

CURRENT PRACTICE

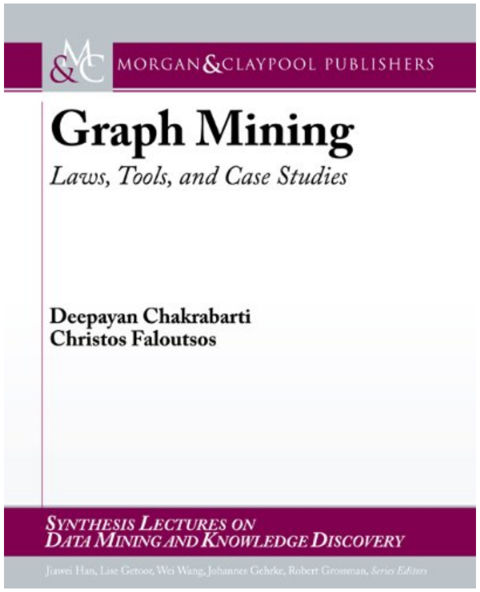
OUR METHOD



Carnegie Mellon

# References

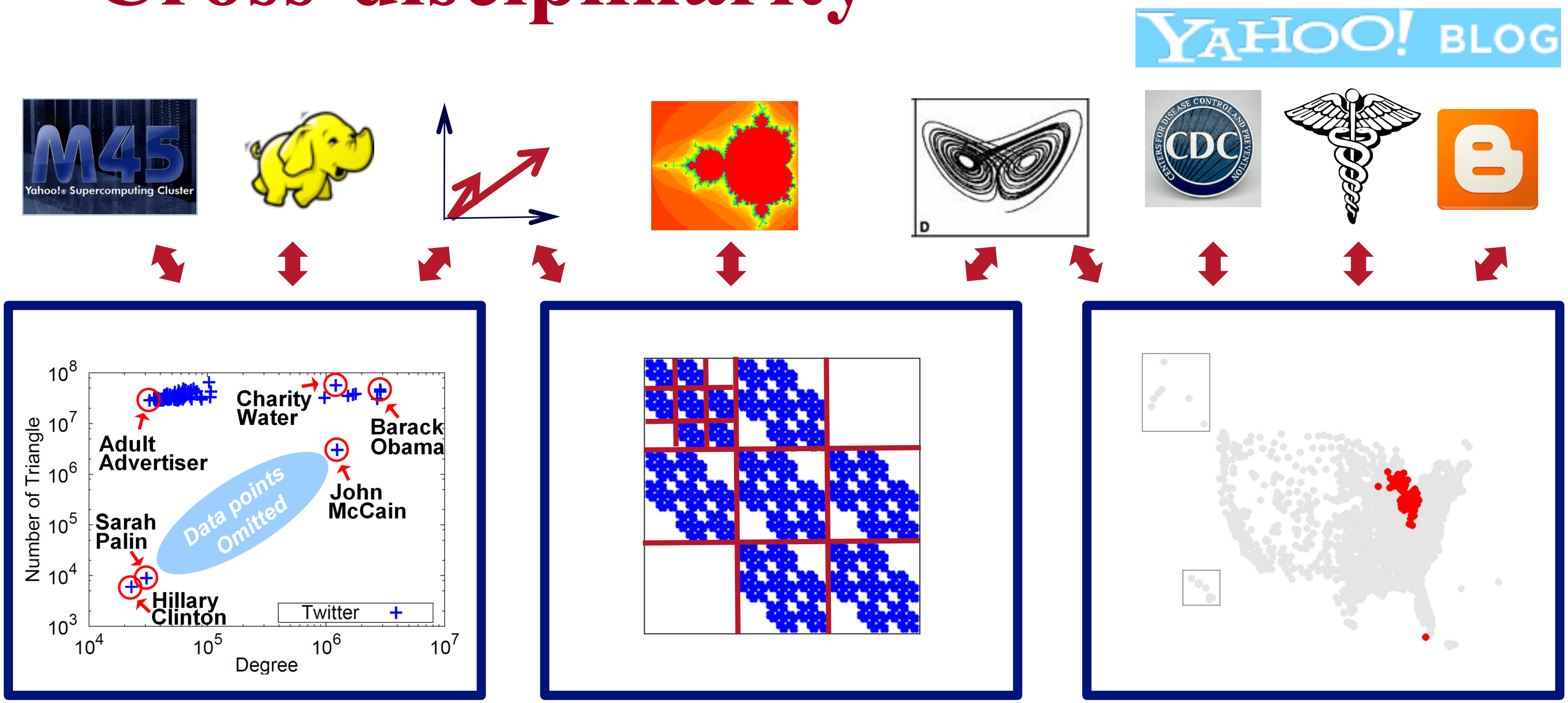
- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
- http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006



Carnegie Mellon

# TAKE HOME MESSAGE:

## Cross-disciplinarity



# QUESTIONS?

## Cross-disciplinarity