Carnegie Mellon

Mining Billion-Scale Graphs: Patterns and Algorithms

Christos Faloutsos and U Kang

CMU

Part 1: Patterns

Complementary to tutorial: *Mining Billion-Scale Graphs: Systems and Implementations*: Haixun Wang et al

Part 1:
Patterns and tools

Our goal:

Open source system for mining huge graphs:

PEGASUS project (PEta GrAph mining System)

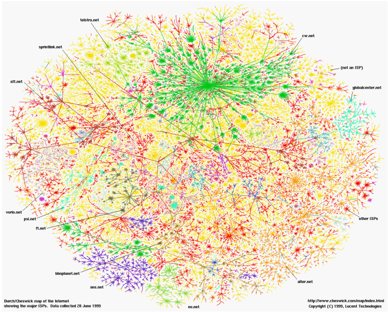
- www.cs.cmu.edu/~pegasus
- code and papers

PROJECT PEGASUS

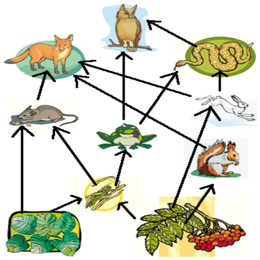
Outline

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

Graphs - why should we care?



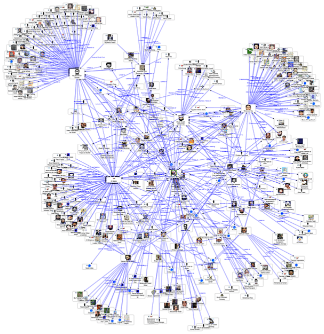
Internet Map
[cheswick.com]



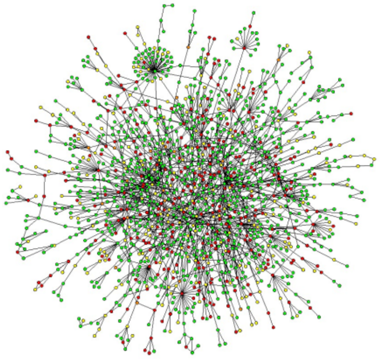
Food Web
[biologycorner.com]

facebook

twitter



Friendship Network
[fmsag.com]

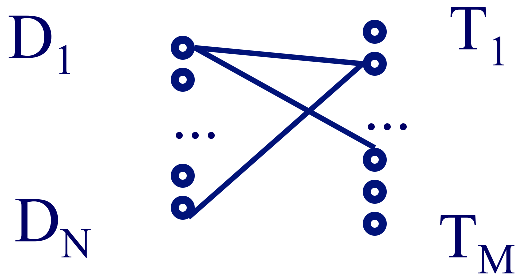


Protein Interactions
[bordalierinstitute.com]

• • •

Graphs - why should we care?

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



- web: hyper-text graph

- ... and more:

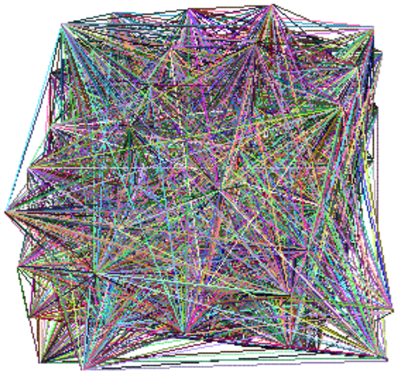
Graphs - why should we care?

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

Outline

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

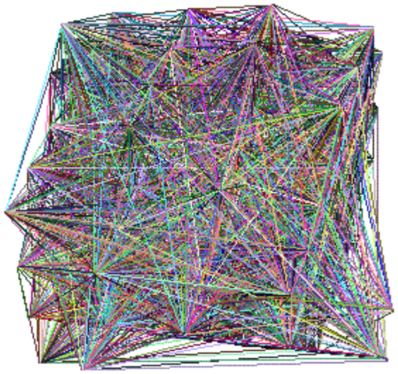
Problem #1 - network and graph mining



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

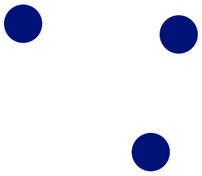
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?

Problem #1 - network and graph mining

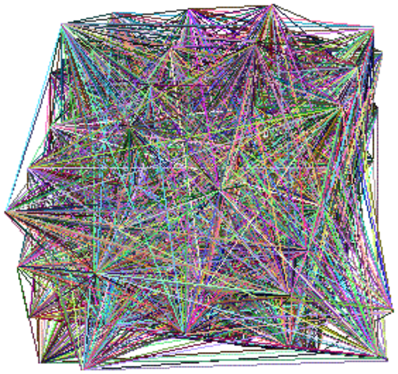


- How does the Internet look like?
- How does FaceBook look like?

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

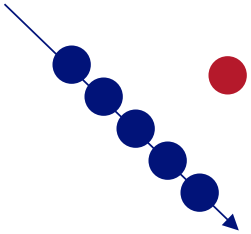


Problem #1 - network and graph mining



- 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…



Graph mining

- Are real graphs random?

Laws and patterns

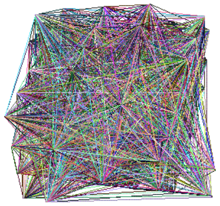
- Are real graphs random?
- A: NO!!
  - Diameter
  - in- and out- degree distributions
  - other (surprising) patterns

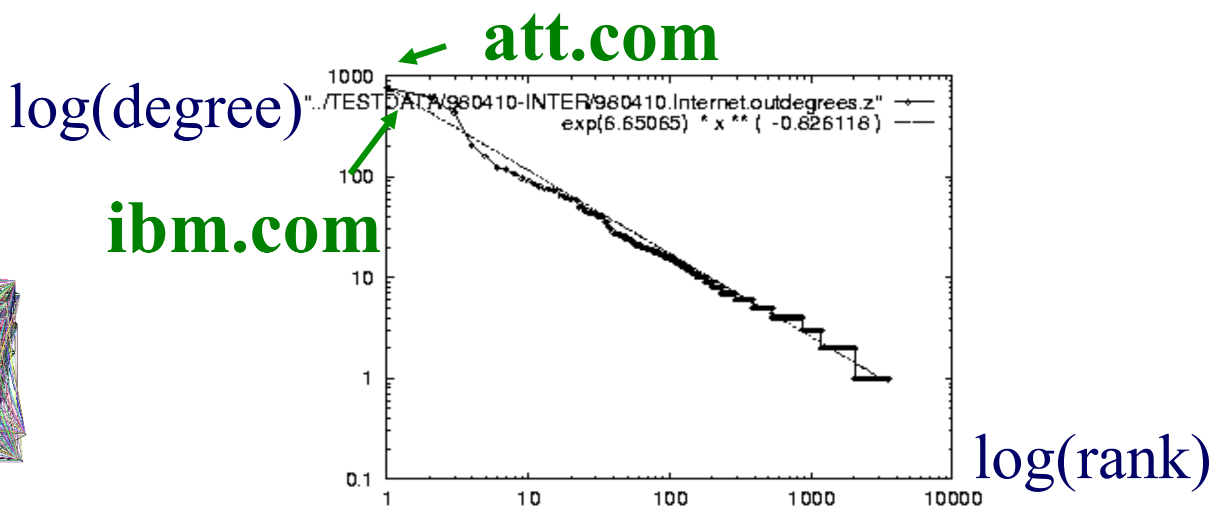
- So, let’s look at the data

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

internet domains



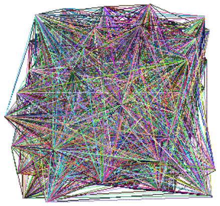


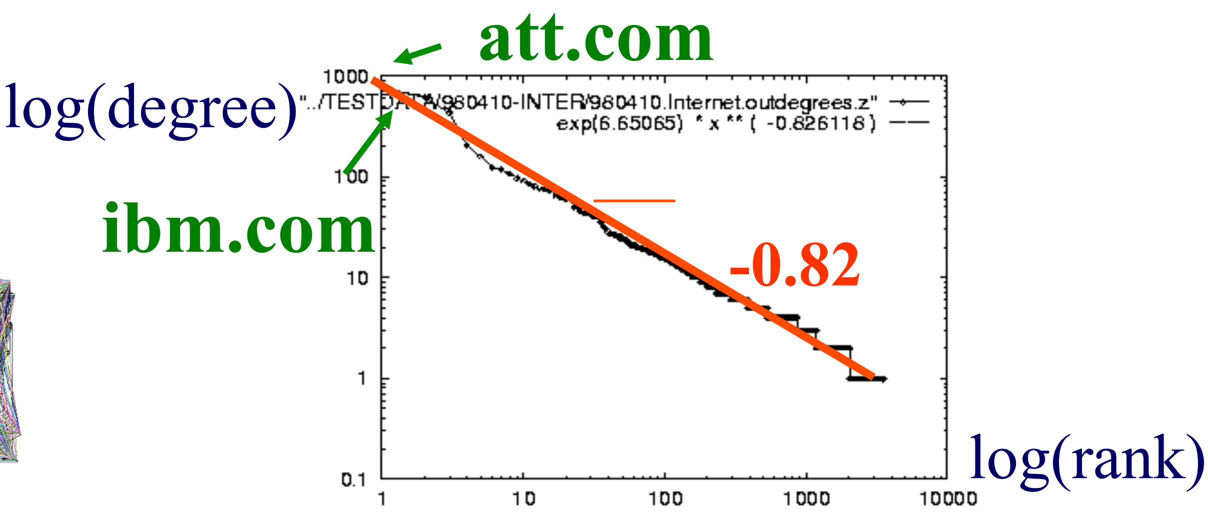
Carnegie Mellon

Solution# S.1

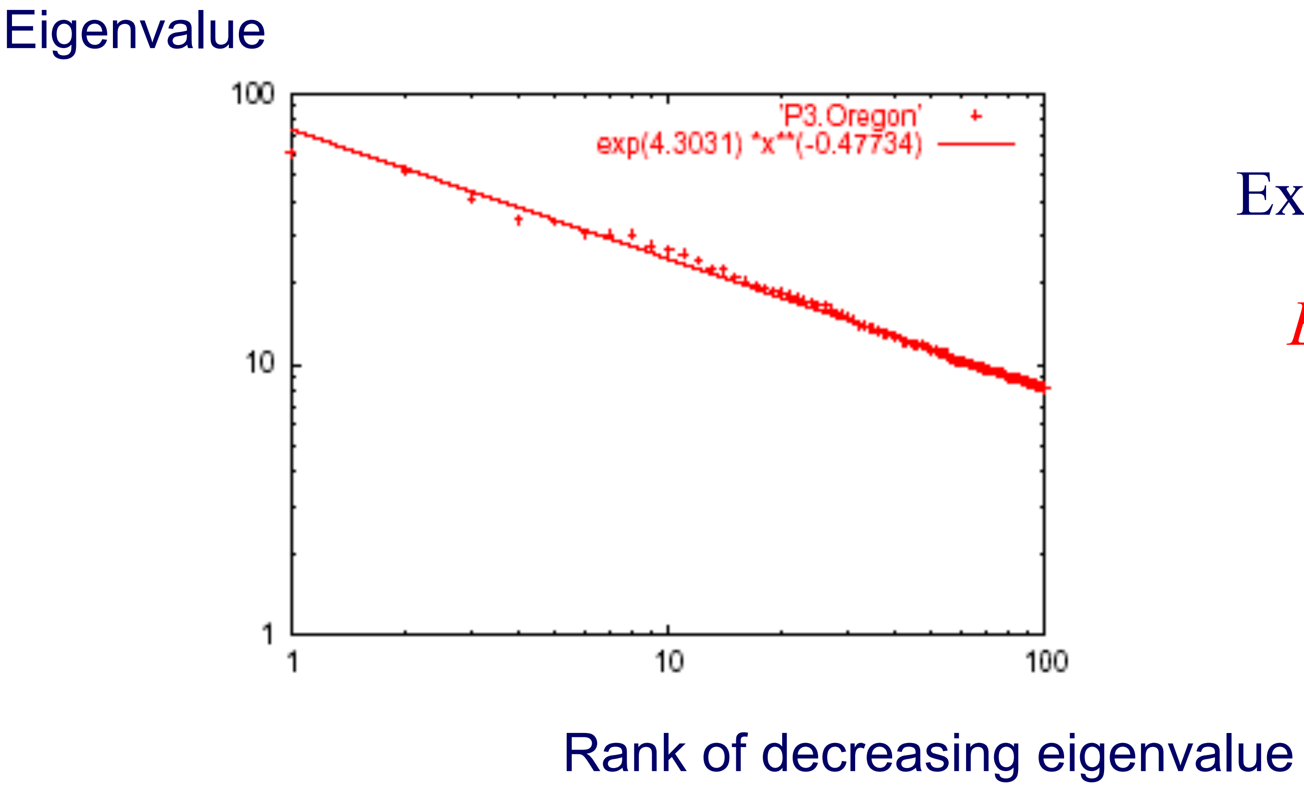
- Power law in the degree distribution [SIGCOMM99]

internet domains





Solution# S.2: Eigen Exponent *E*



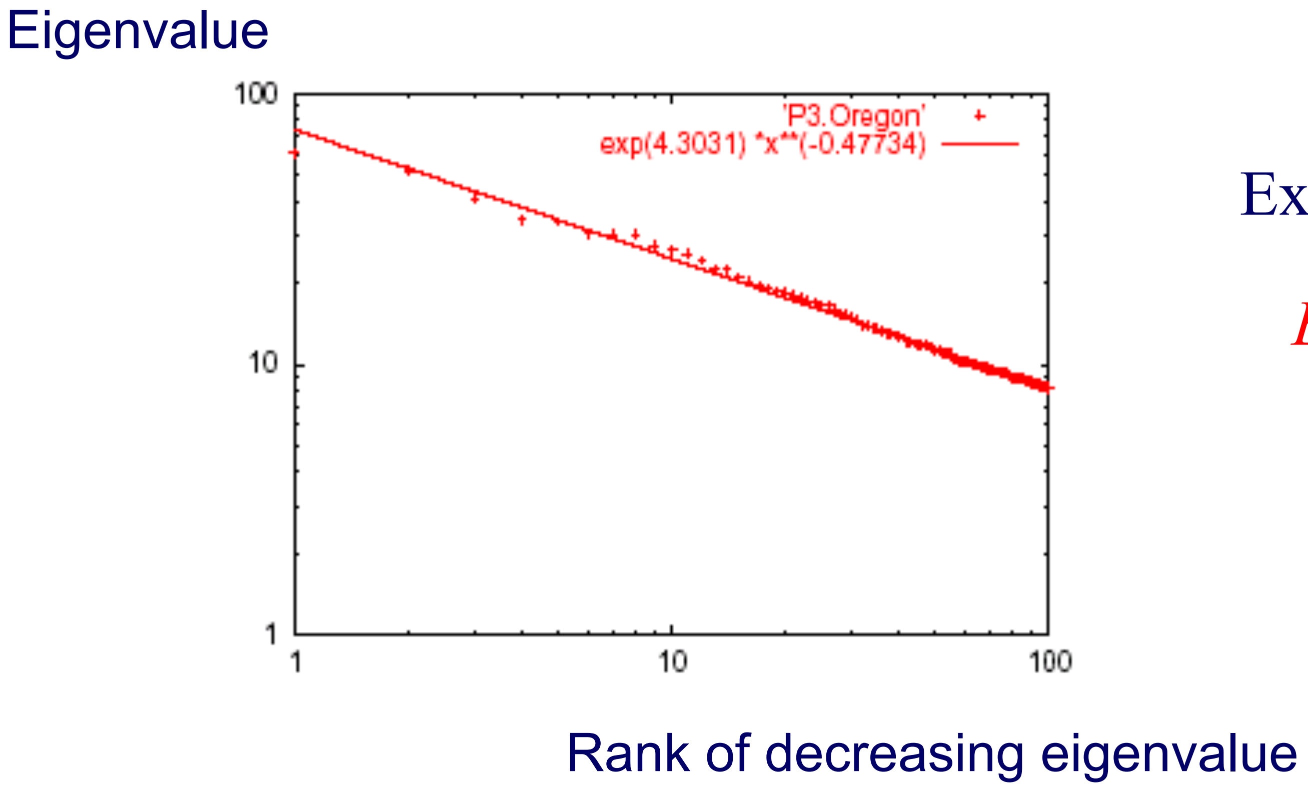
Exponent = slope

E = -0.48

May 2001

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

Solution# S.2: Eigen Exponent *E*



Exponent = slope

$E = -0.48$

May 2001

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

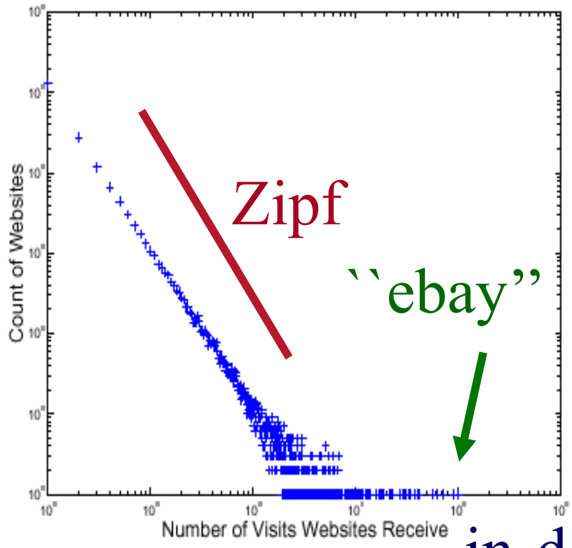
But:

How about graphs from other domains?

More power laws:

- web hit counts [w/ A. Montgomery]

Web Site Traffic

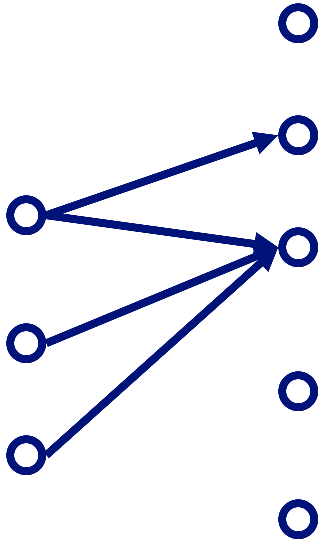


Count (log scale)

Zipf

``ebay''

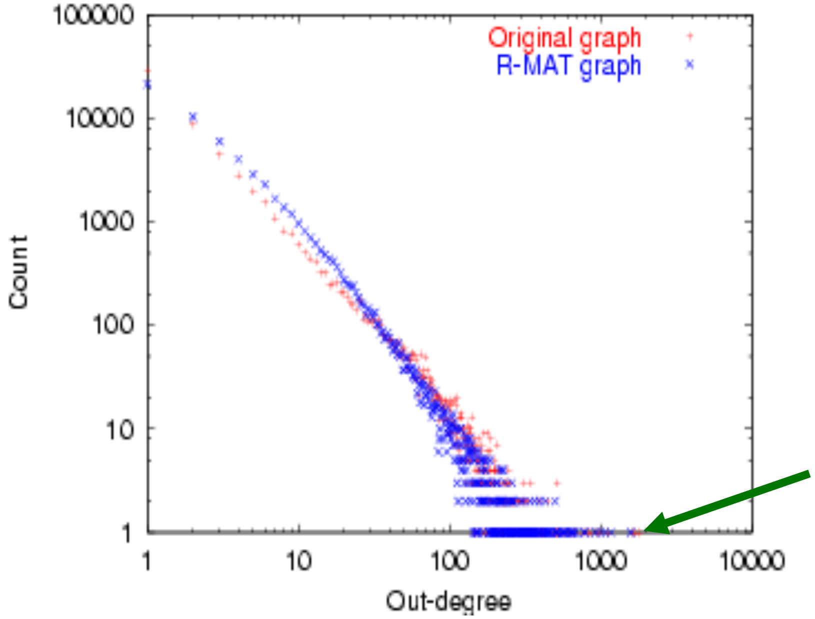
in-degree (log scale)



epinions.com

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

count



trusts-2000-people user

(out) degree

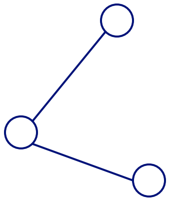
And numerous more

- # 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

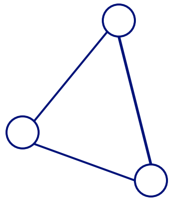
- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - ➡ triangles
    - cliques
  - Weighted graphs
  - Time evolving graphs
- Problem#2: Tools

Solution# S.3: Triangle 'Laws'



- Real social networks have a lot of triangles

Solution# S.3: Triangle ‘Laws’

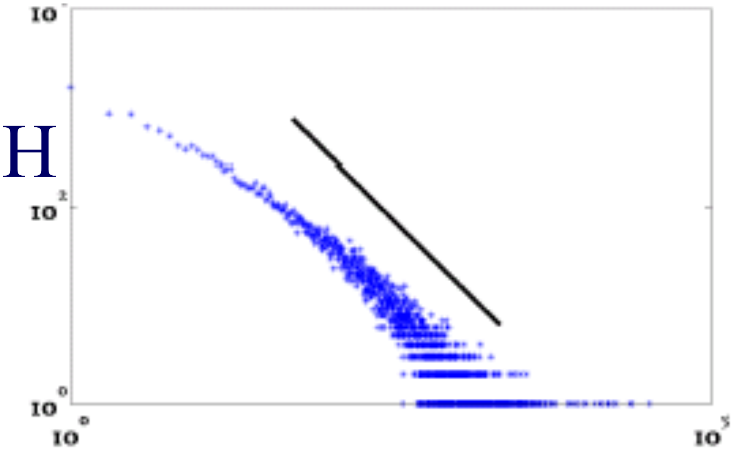


- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

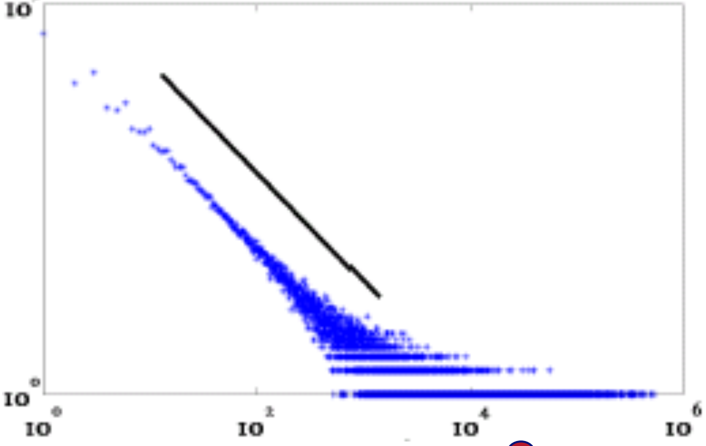
Carnegie Mellon

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

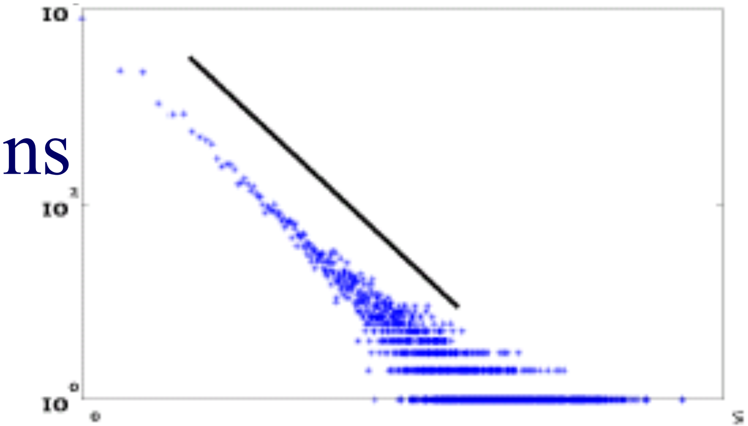
HEP-TH

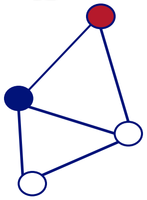


ASN



Epinions





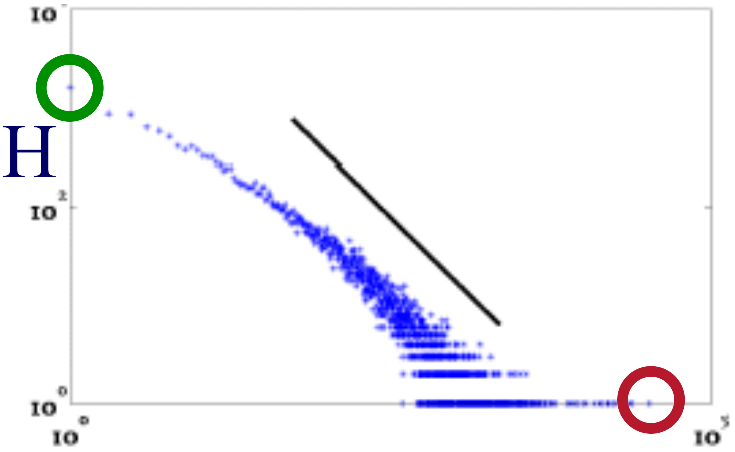
X-axis: # of participating triangles

Y: count (~ pdf)

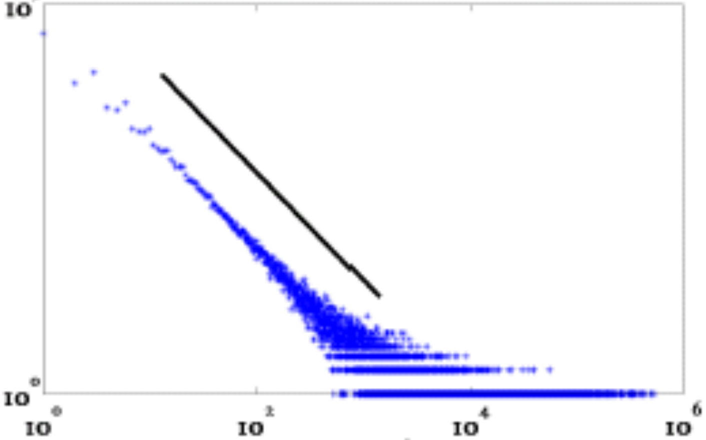
Triangle Law: #S.3

[Tsourakakis ICDM 2008]

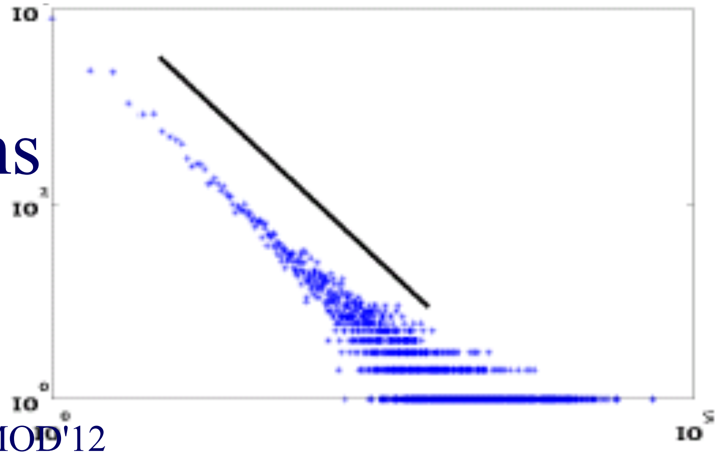
HEP-TH

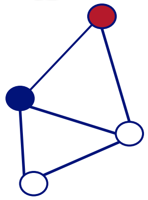


ASN



Epinions





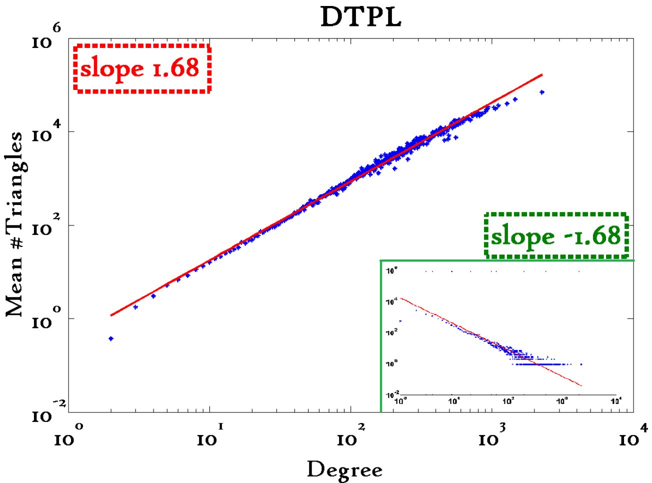
X-axis: # of participating triangles

Y: count (~ pdf)

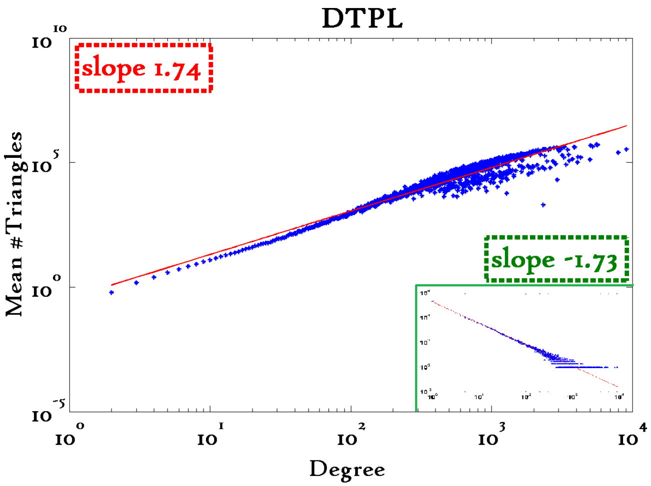
Triangle Law: #S.4

[Tsourakakis ICDM 2008]

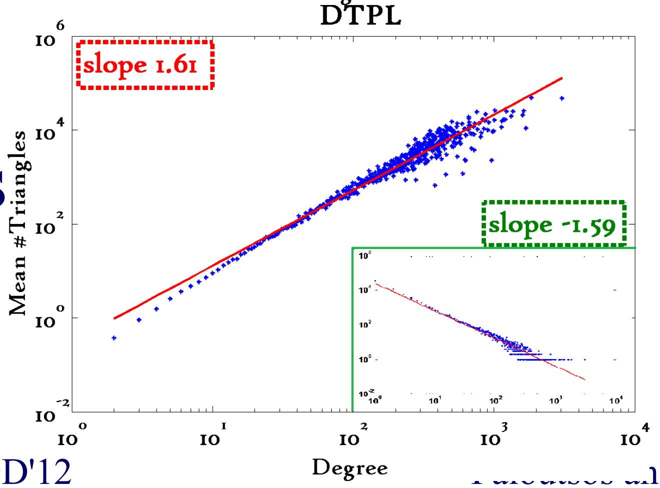
Reuters



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?

details

Triangle Law: Computations

[Tsourakakis ICDM 2008]

But: triangles are expensive to compute

(3-way join; several approx. algos)

Q: Can we do that quickly?

A: Yes!

#triangles = 1/6 Sum (λ_i^3)

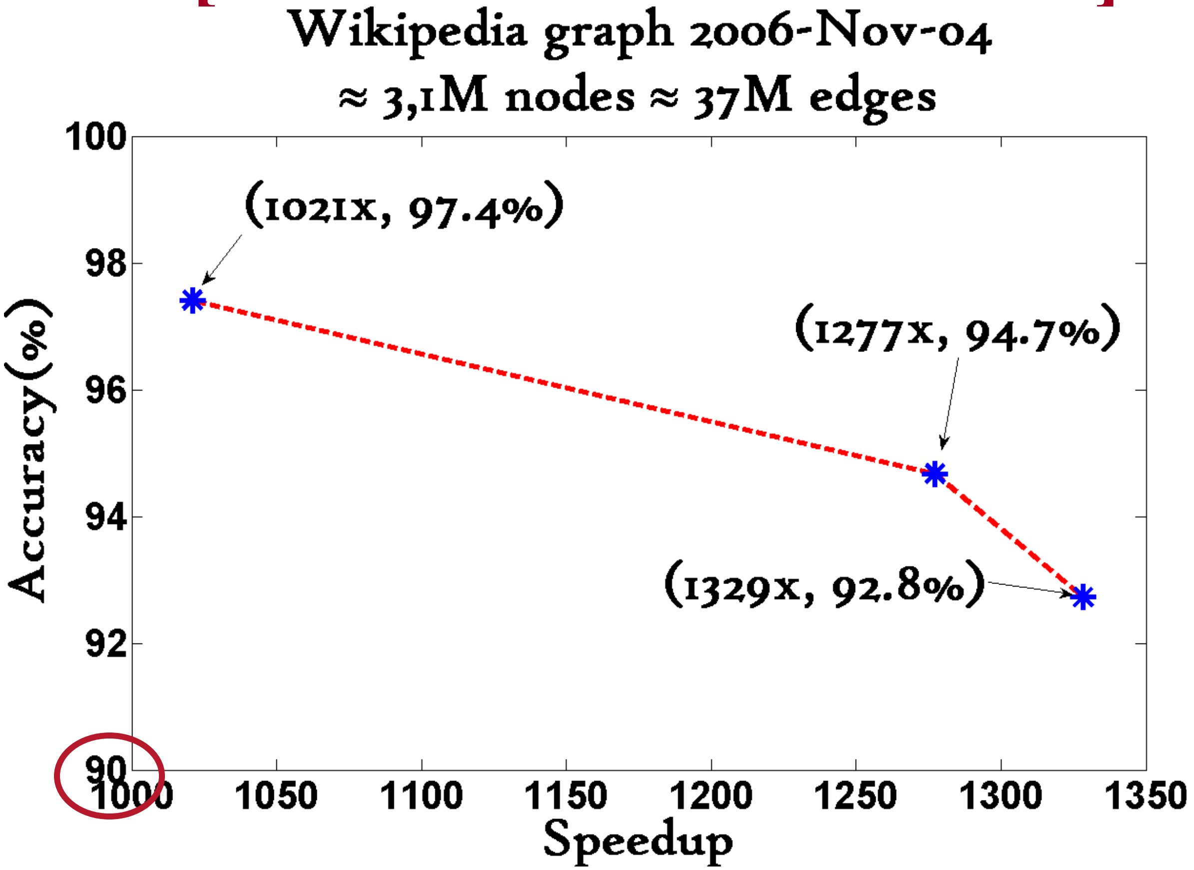
(and, because of skewness (S2) ,
we only need the top few eigenvalues!)

Carnegie Mellon

details

Triangle Law: Computations

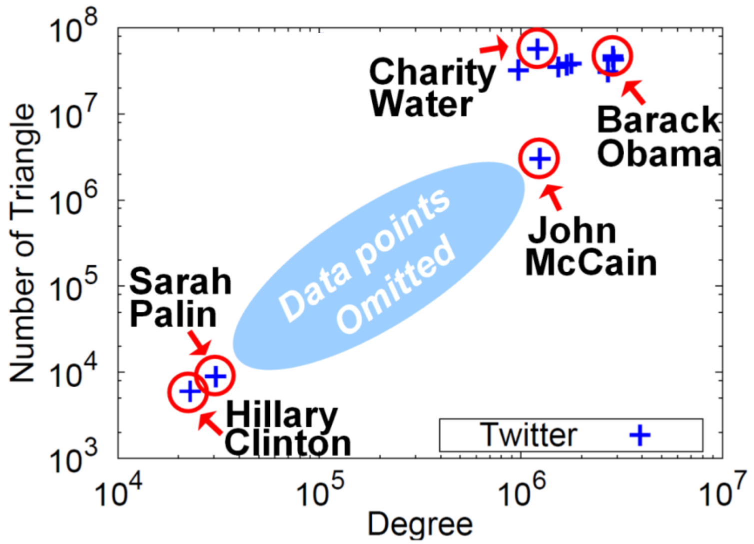
[Tsourakakis ICDM 2008]



1000x+ speed-up, >90% accuracy

Triangle for Anomaly Detection

- Triangle counting in Twitter social network

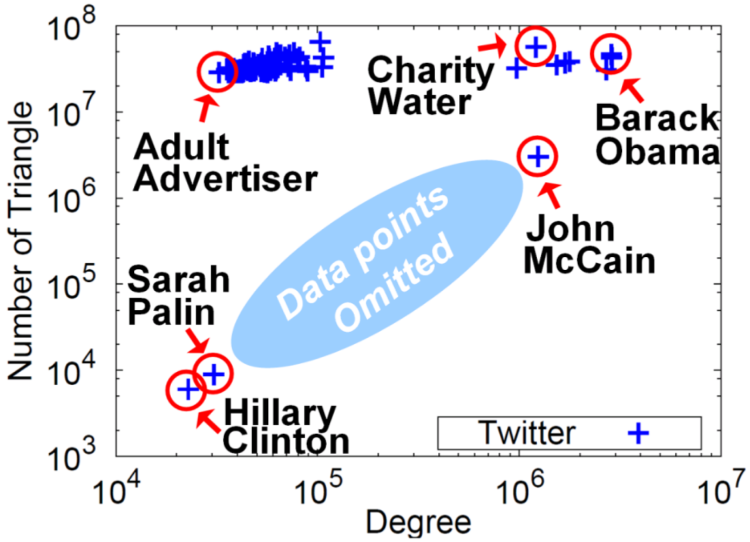


[Twitter 2009;
~ 60 million nodes
~ 3 billion edges]

- U.S. politicians: moderate number of triangles vs. degree

Triangle for Anomaly Detection

- Triangle counting in Twitter social network



[Twitter 2009;
~ 60 million nodes
~ 3 billion edges]

- U.S. politicians: moderate number of triangles vs. degree
  - Adult sites: very large number of triangles vs. degree

Outline

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

Observations on weighted graphs?

- A: yes - even more 'laws'!

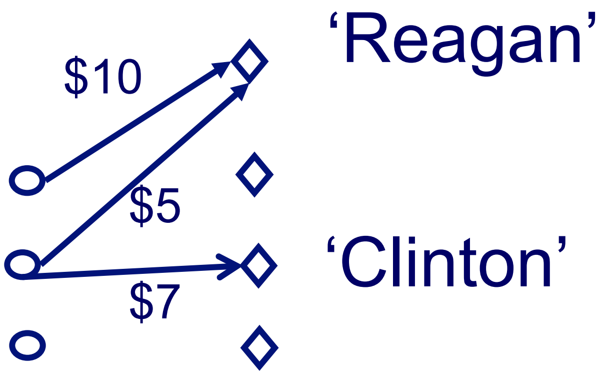
M. McGlohon, L. Akoglu, and C. Faloutsos
Weighted Graphs and Disconnected Components: Patterns and a Generator.
SIG-KDD 2008

Observation W.1: Fortification

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

Observation W.1: Fortification

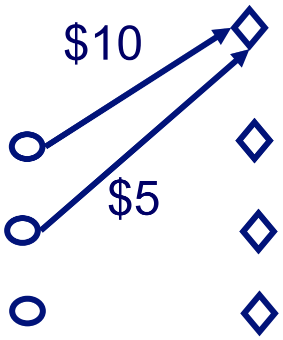
More donors, more $?



Observation W.1: fortification: Snapshot Power Law

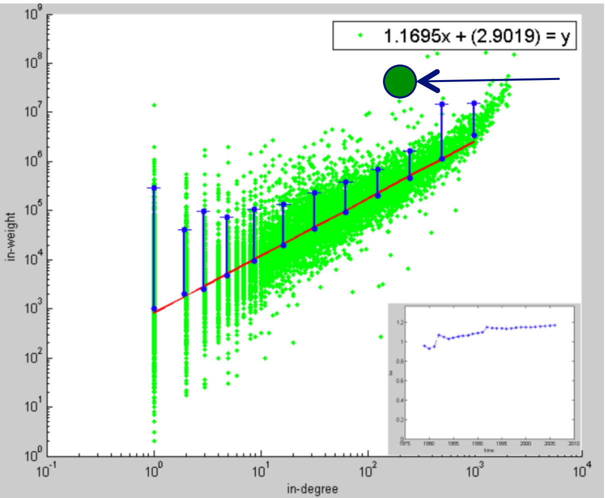
- Weight: super-linear on in-degree
- exponent ‘iw’: 1.01 < iw < 1.26

More donors, even more $



In-weights ($)

Orgs-Candidates



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

Edges (# donors)

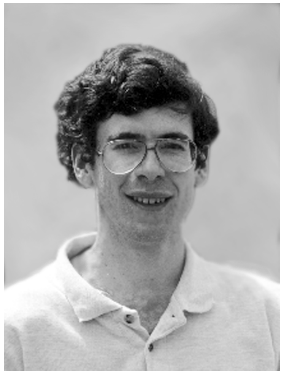
Outline

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

Problem: Time evolution

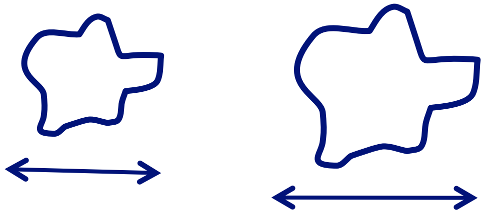
- with Jure Leskovec (CMU -> Stanford)

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



T.1 Evolution of the Diameter

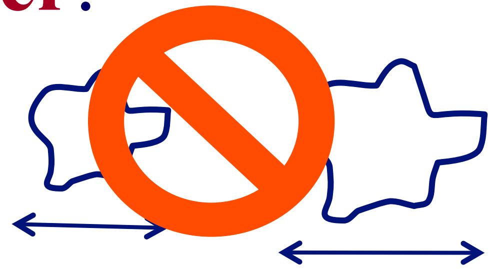
- Prior work on Power Law graphs hints at **slowly growing diameter**:
  - – diameter ~ O(log N)
  - – diameter ~ O(log log N)



- 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)

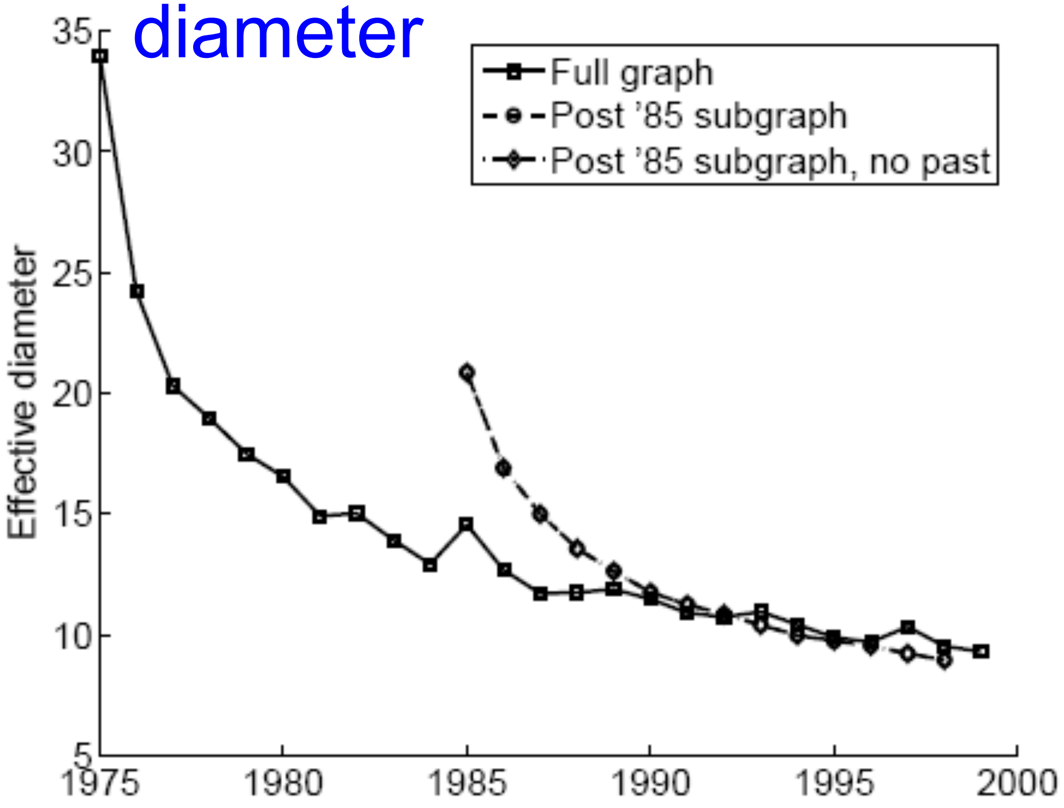


- What is happening in real data?
- Diameter **shrinks** over time

T.1 Diameter – "Patents"

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

diameter



time [years]

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

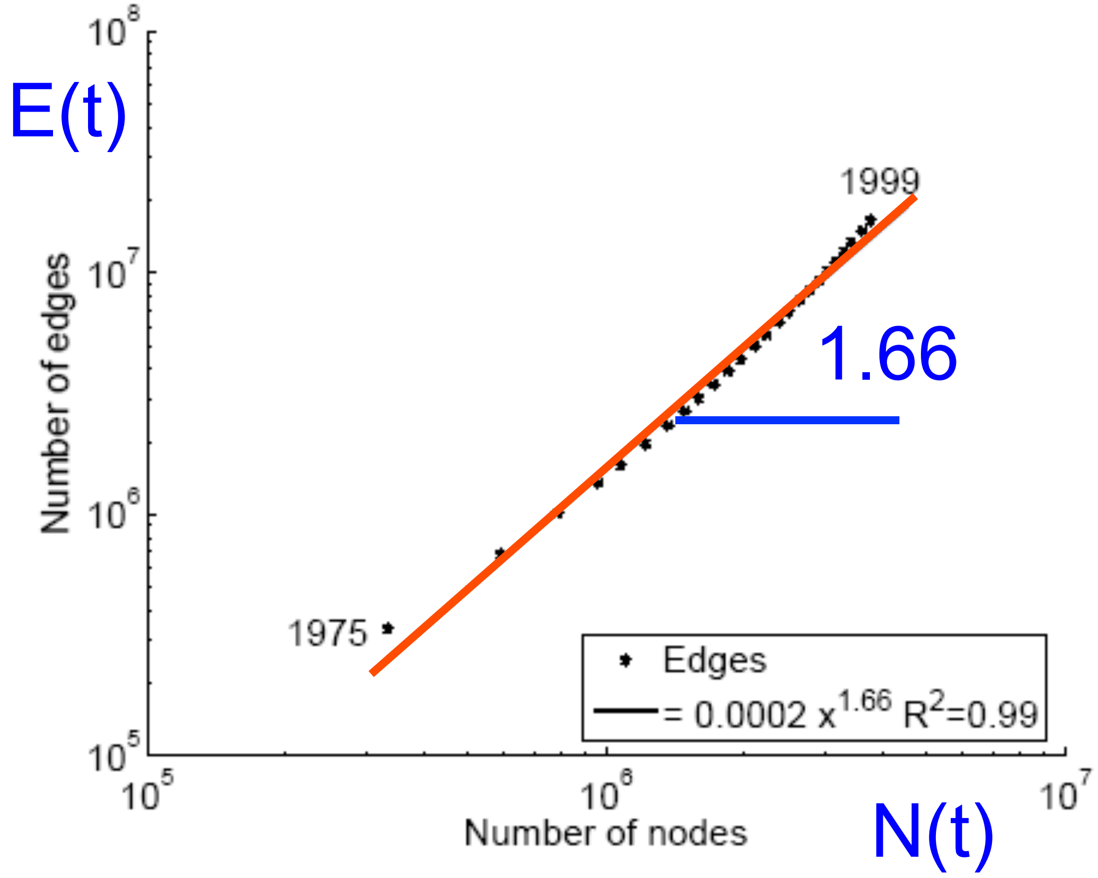
- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

  $N(t+1) = 2 * N(t)$
- 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
- Problem#1: Patterns in graphs
  - Static graphs
  - Weighted graphs
  - ➡ Time evolving graphs
- Problem#2: Tools
- …

More on Time-evolving graphs

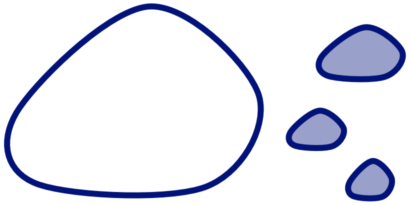
M. McGlohon, L. Akoglu, and C. Faloutsos
Weighted Graphs and Disconnected Components: Patterns and a Generator.
SIG-KDD 2008

Observation T.3: NLCC behavior

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

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

- Do they continue to grow in size?
- or do they shrink?
- or stabilize?



Observation T.3: NLCC behavior

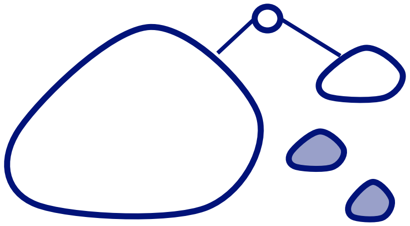
Q: How do NLCC's emerge and join with the GCC?

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

– Do they continue to grow in size?

– or do they shrink?

– or stabilize?

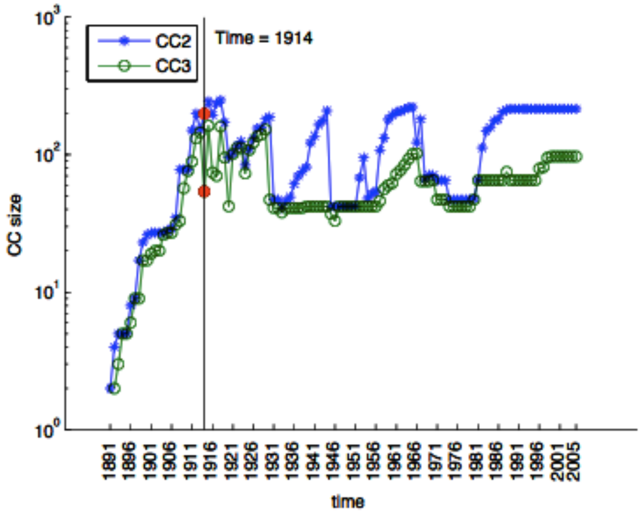


Observation T.3: NLCC behavior

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

IMDB

CC size



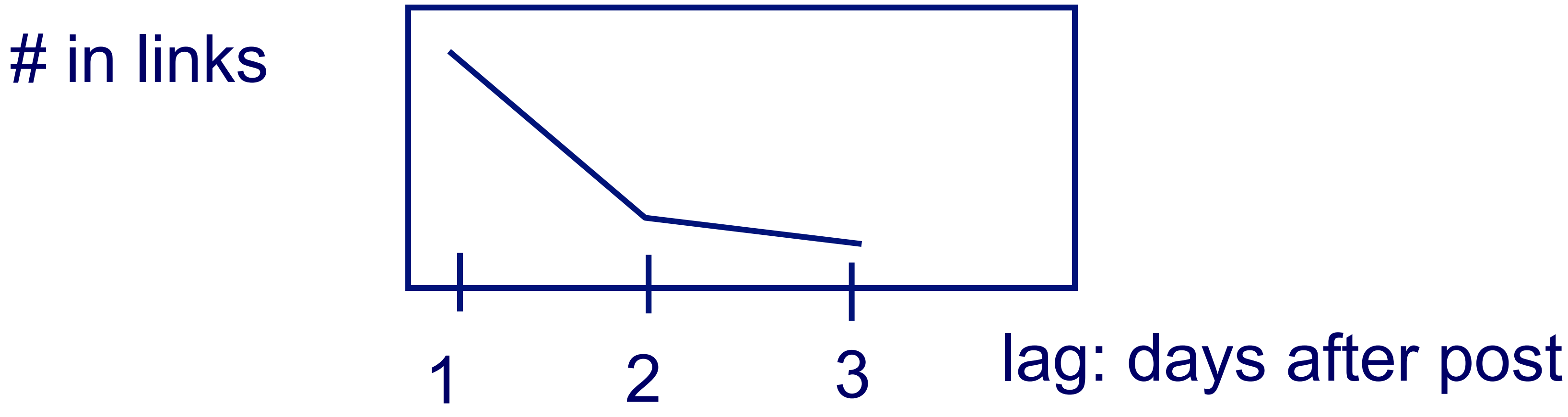
Time-stamp

Timing for Blogs

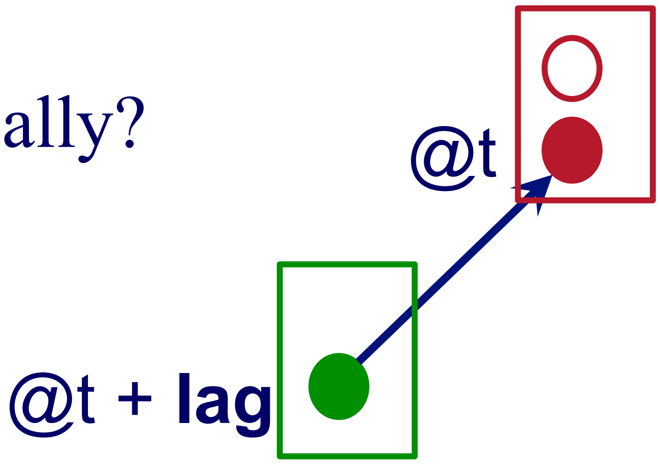
- with Mary McGlohon (CMU->google)
- Jure Leskovec (CMU->Stanford)
- Natalie Glance (now at Google)
- Mat Hurst (now at MSR)

[SDM'07]

T.4 : popularity over time



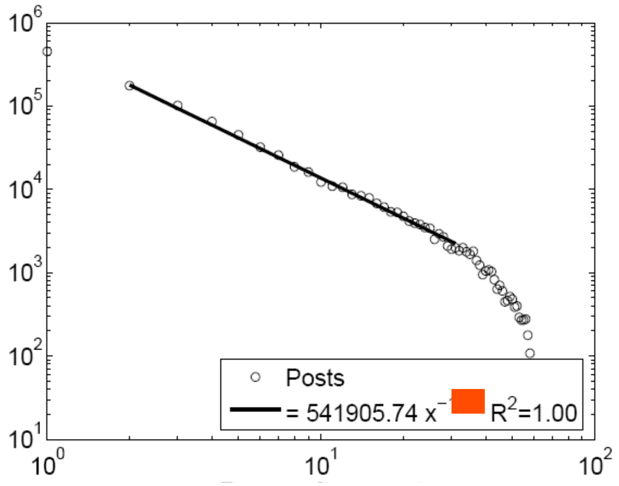
Post popularity drops-off – exponentially?



Carnegie Mellon

T.4 : popularity over time

in links
(**log**)



days after post
(**log**)

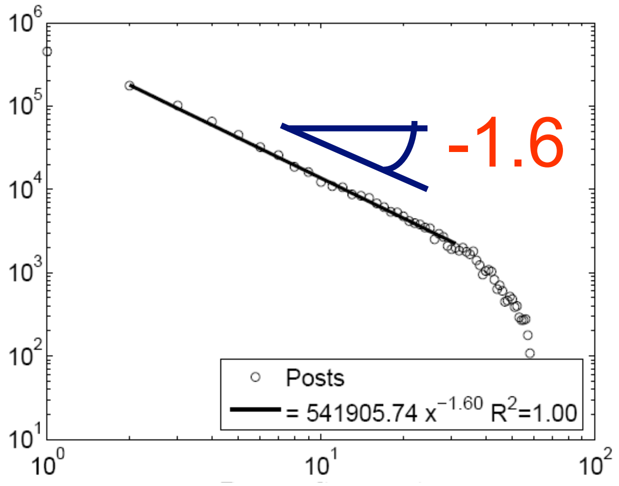
Post popularity drops-off – exponentially?

POWER LAW!

Exponent?

T.4 : popularity over time

in links
(**log**)



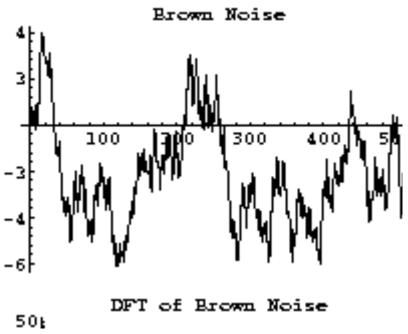
days after post
(**log**)

Post popularity drops-off – exponentially?

POWER LAW!

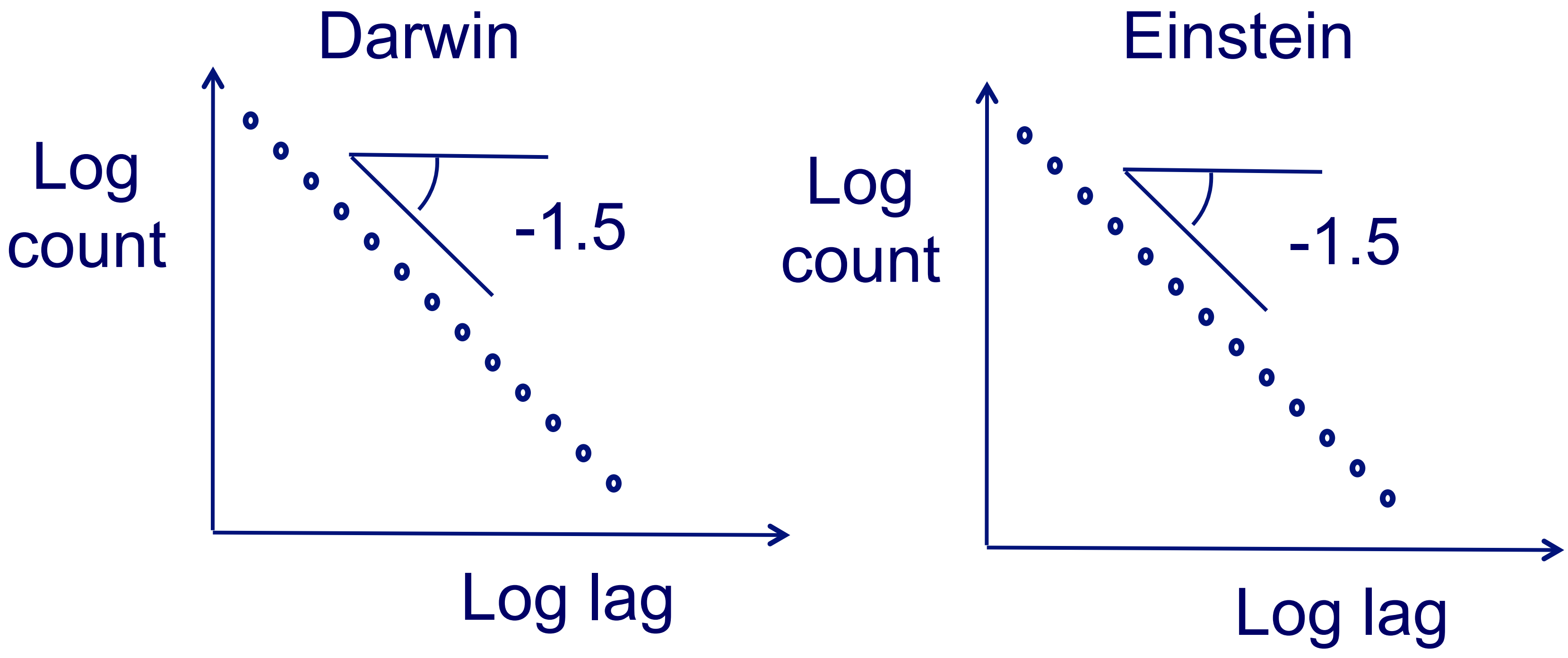
Exponent? -1.6

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



-1.5 slope

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



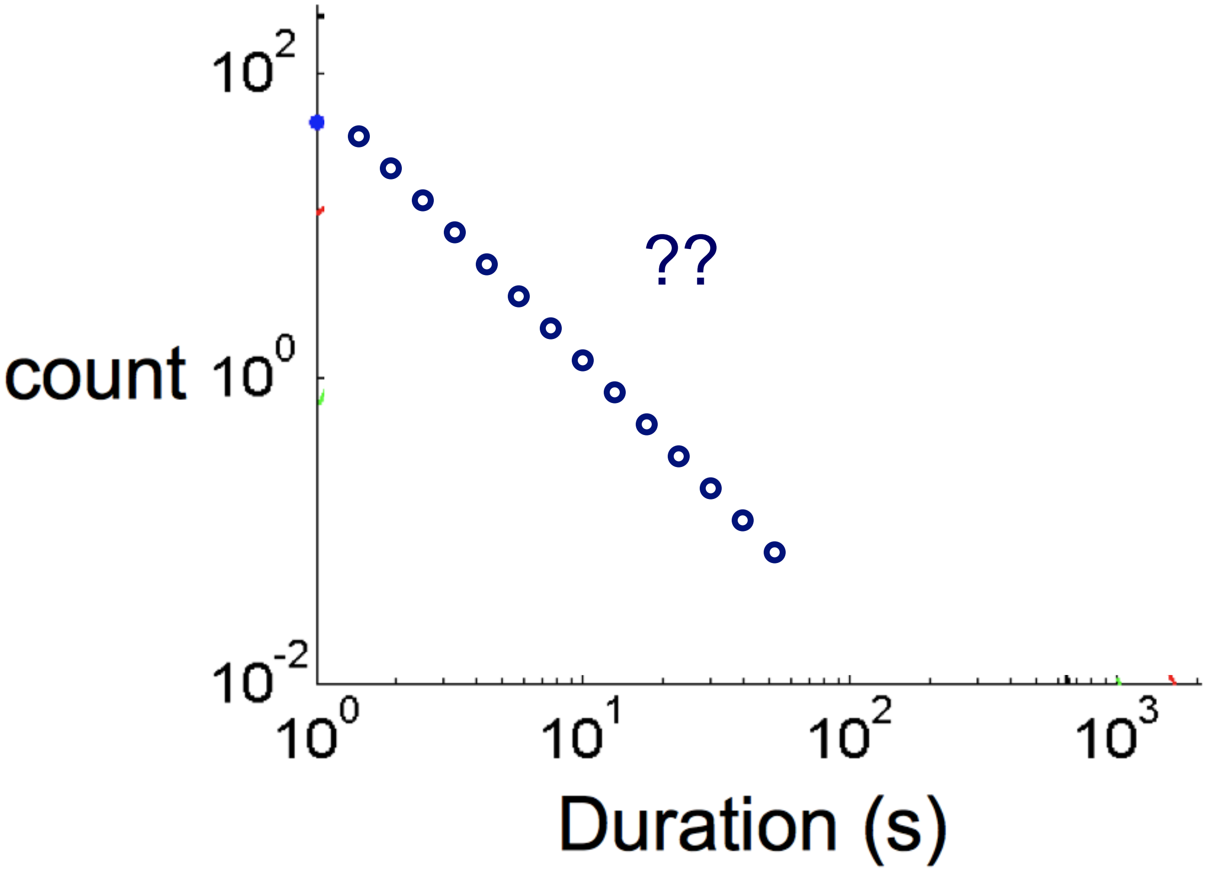
T.5: duration of phonecalls

Surprising Patterns for the Call Duration Distribution of Mobile Phone Users

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

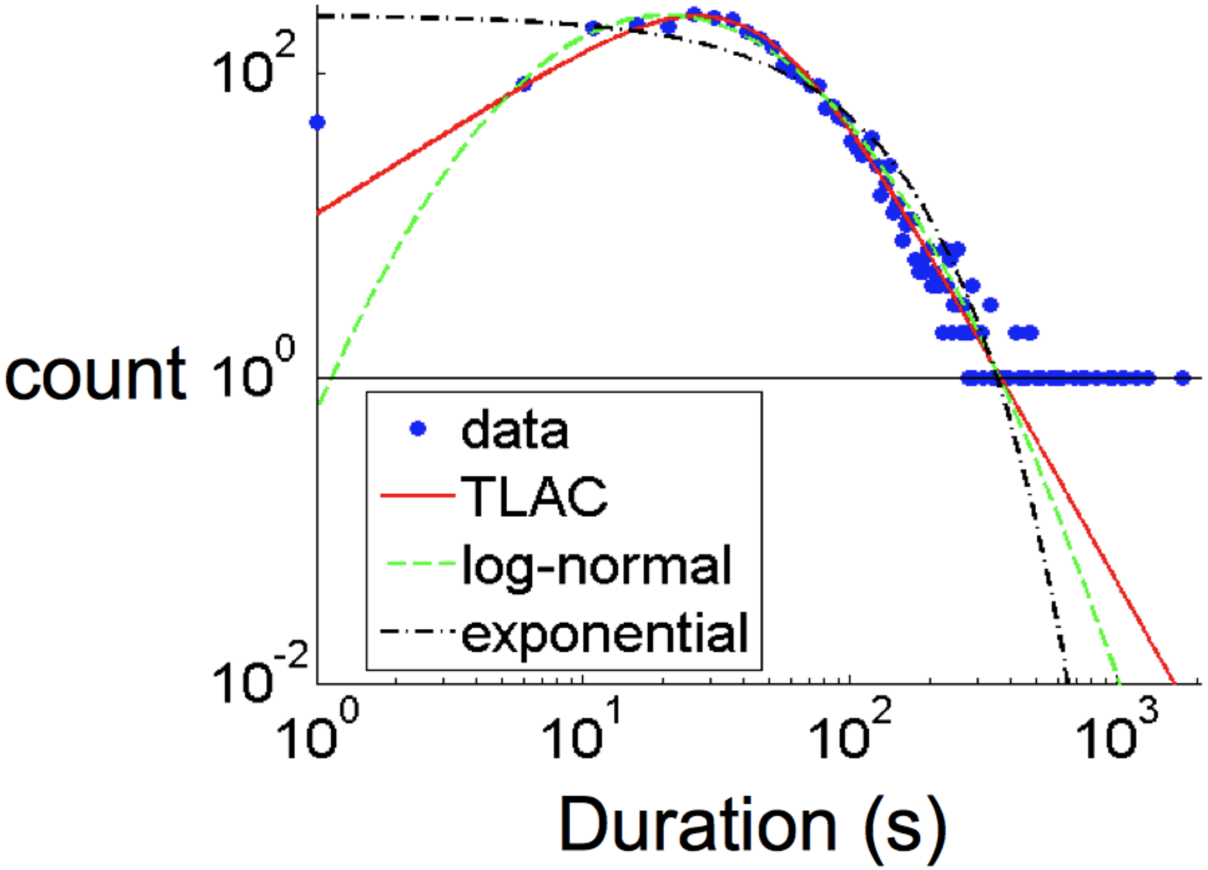
PKDD 2010

Probably, power law (?)



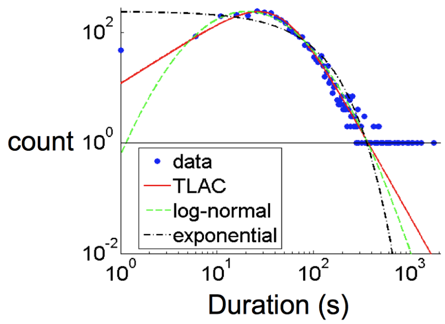
Carnegie Mellon

No Power Law!



‘TLaC: Lazy Contractor’

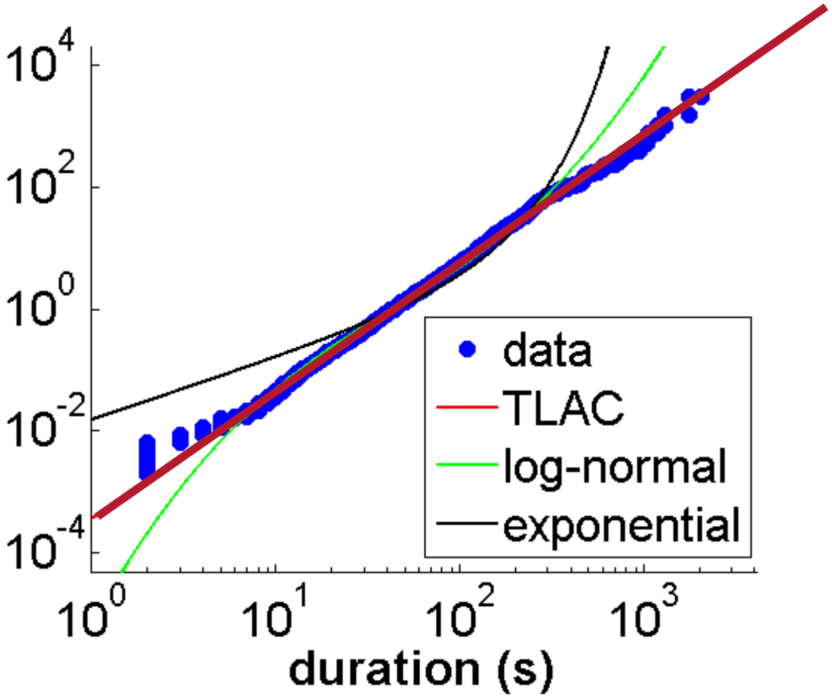
- The longer a task (phonecall) has taken,
- The even longer it will take



Odds ratio=

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

== power law



Data Description

- Data from a private mobile operator of a large city
    - 4 months of data
    - 3.1 million users
    - more than 1 billion phone records

Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- ➡ Problem#2: Tools
  - Oddball
  - SVD
- Problem#3: Algorithms and Scalability
- Conclusions

OddBall: Spotting Anomalies in Weighted Graphs

Leman Akoglu, Mary McGlohon, Christos Faloutsos

Carnegie Mellon University

School of Computer Science

PAKDD 2010, Hyderabad, India

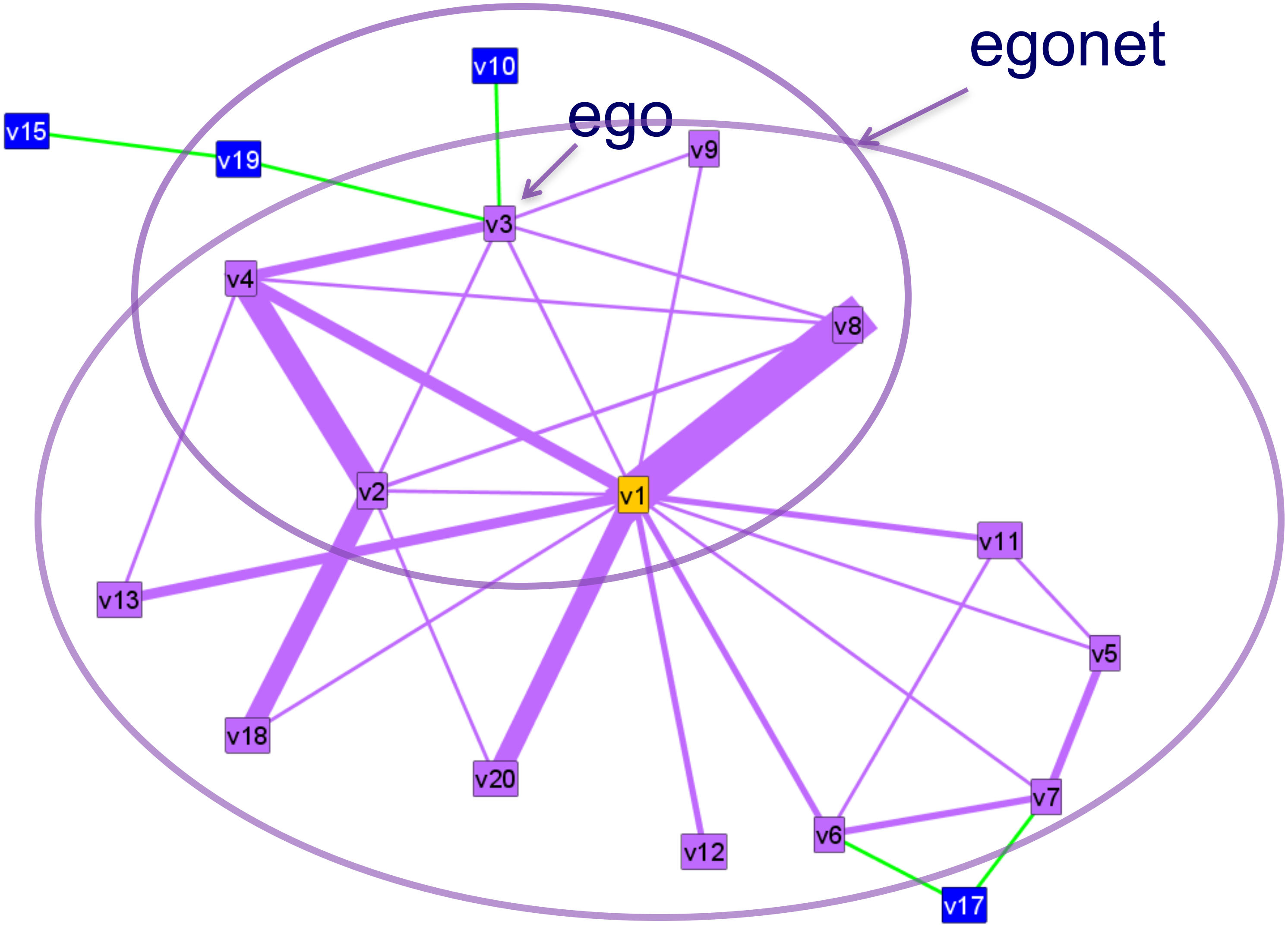
Main idea

For each node,

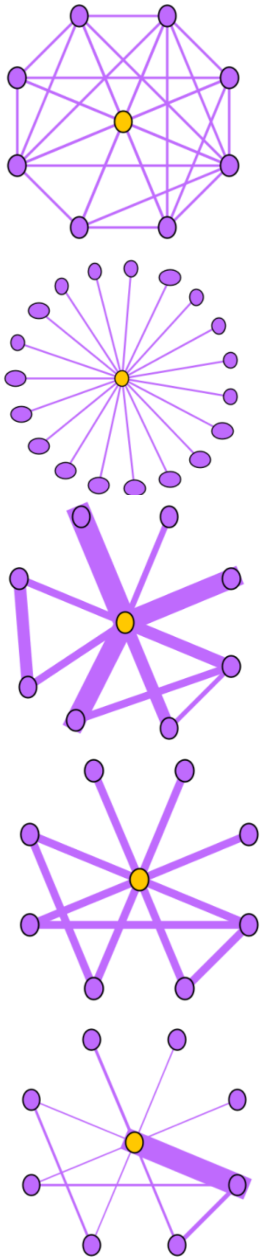
- extract ‘ego-net’ (=1-step-away neighbors)
- Extract features (#edges, total weight, etc etc)
- Compare with the rest of the population

Carnegie Mellon

What is an egonet?

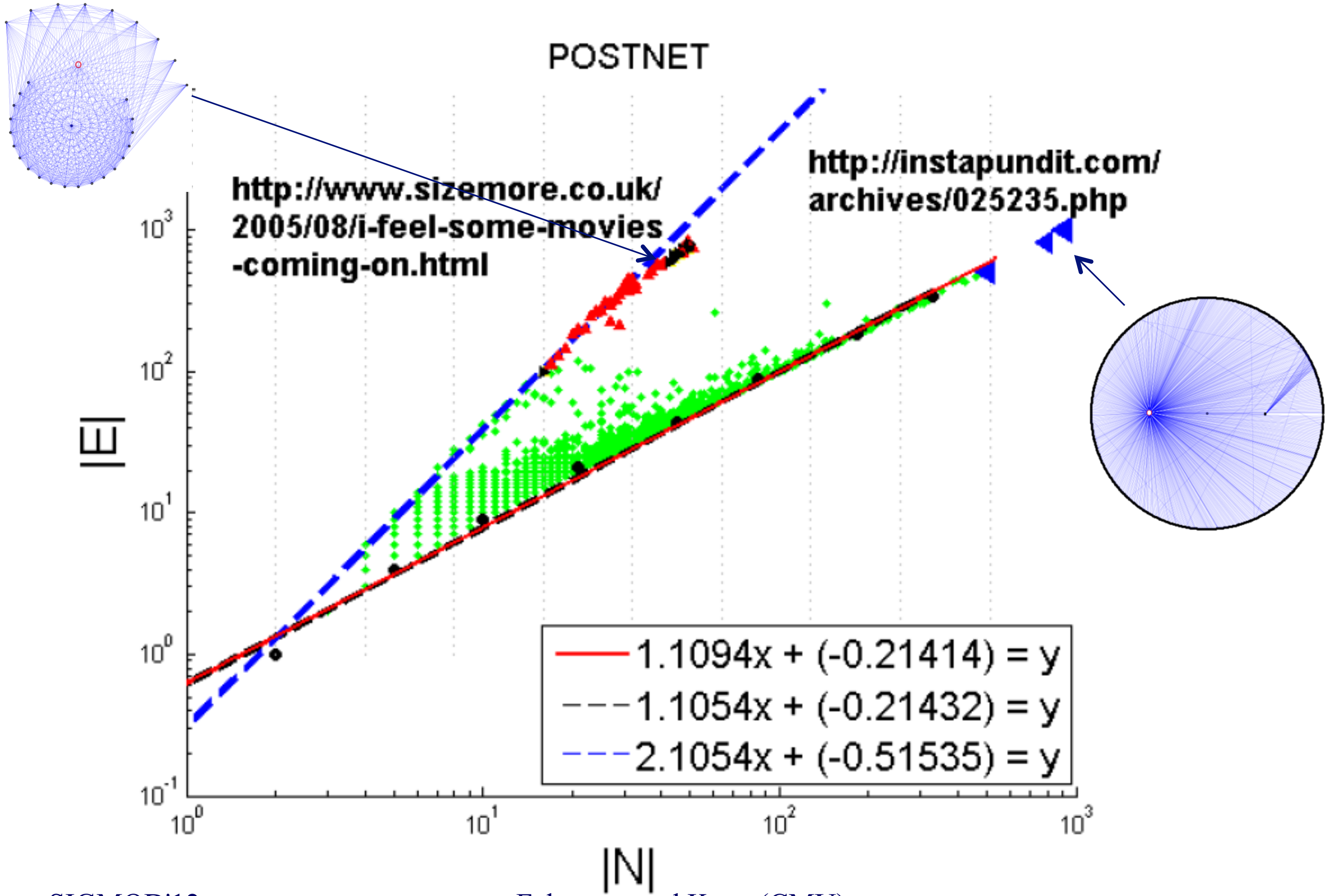


Selected Features

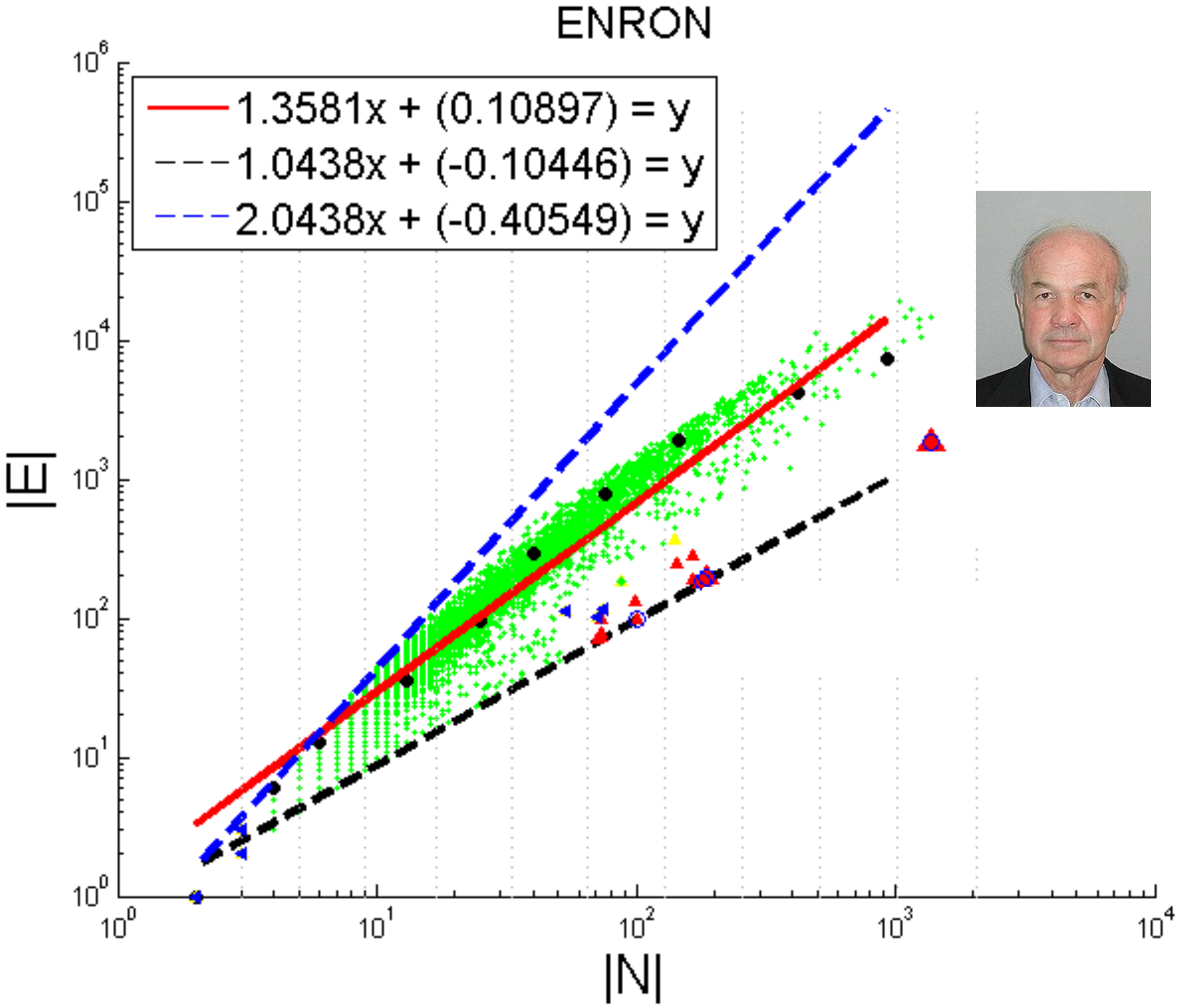


- N_i: number of neighbors (degree) of ego i
- E_i: number of edges in egonet i
- W_i: total weight of egonet i
- $\lambda_{w,i}$: principal eigenvalue of the weighted adjacency matrix of egonet I

Near-Clique/Star



Near-Clique/Star



Outline

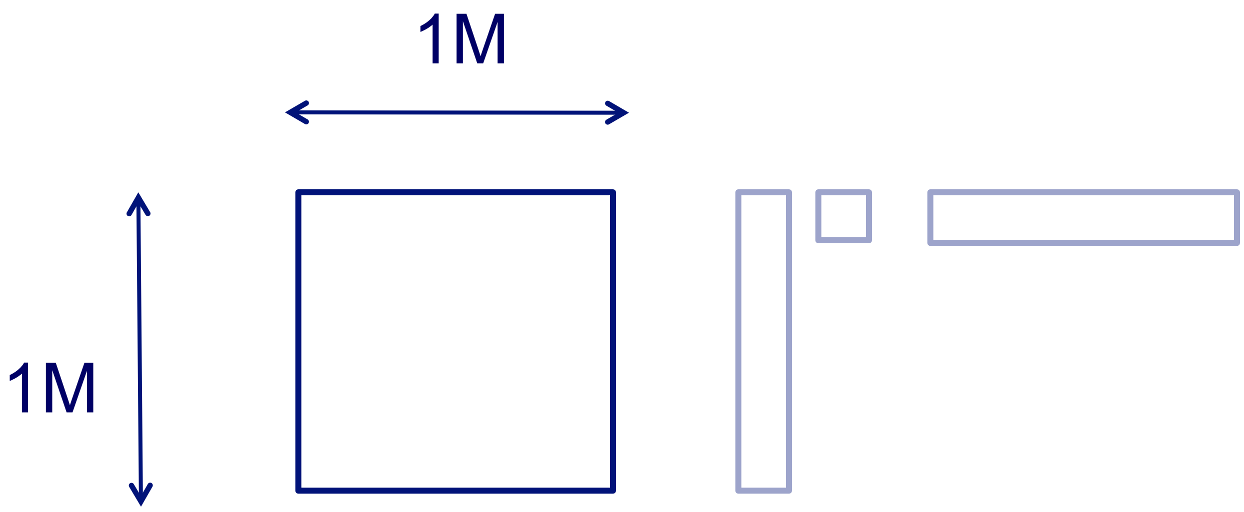
- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Oddball
  - ➡ SVD
- Problem#3: Algorithms and Scalability
- Conclusions

Singular Value Decomposition

- Powerful tool, identical or closely related to
  - Latent Semantic Indexing (LSI)
  - Karhunen-Loeve Transform (KLT)
  - Principal Component Analysis (PCA)
  - …

Motivation

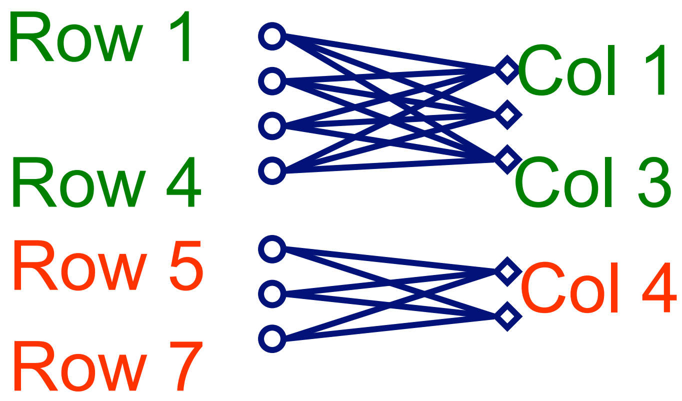
- Who-calls-whom (~1M) – how to visualize/ understand?



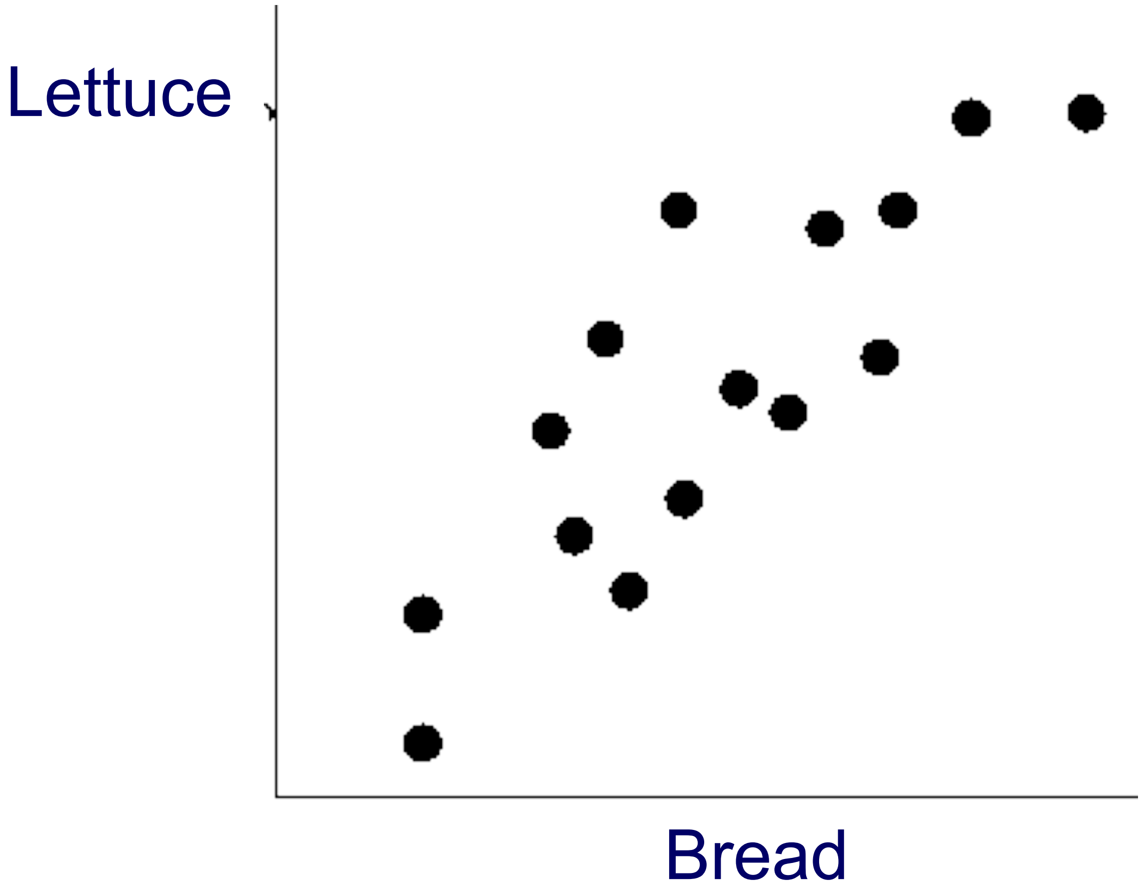
SVD - Motivation

- Customer-product, for recommendation system:

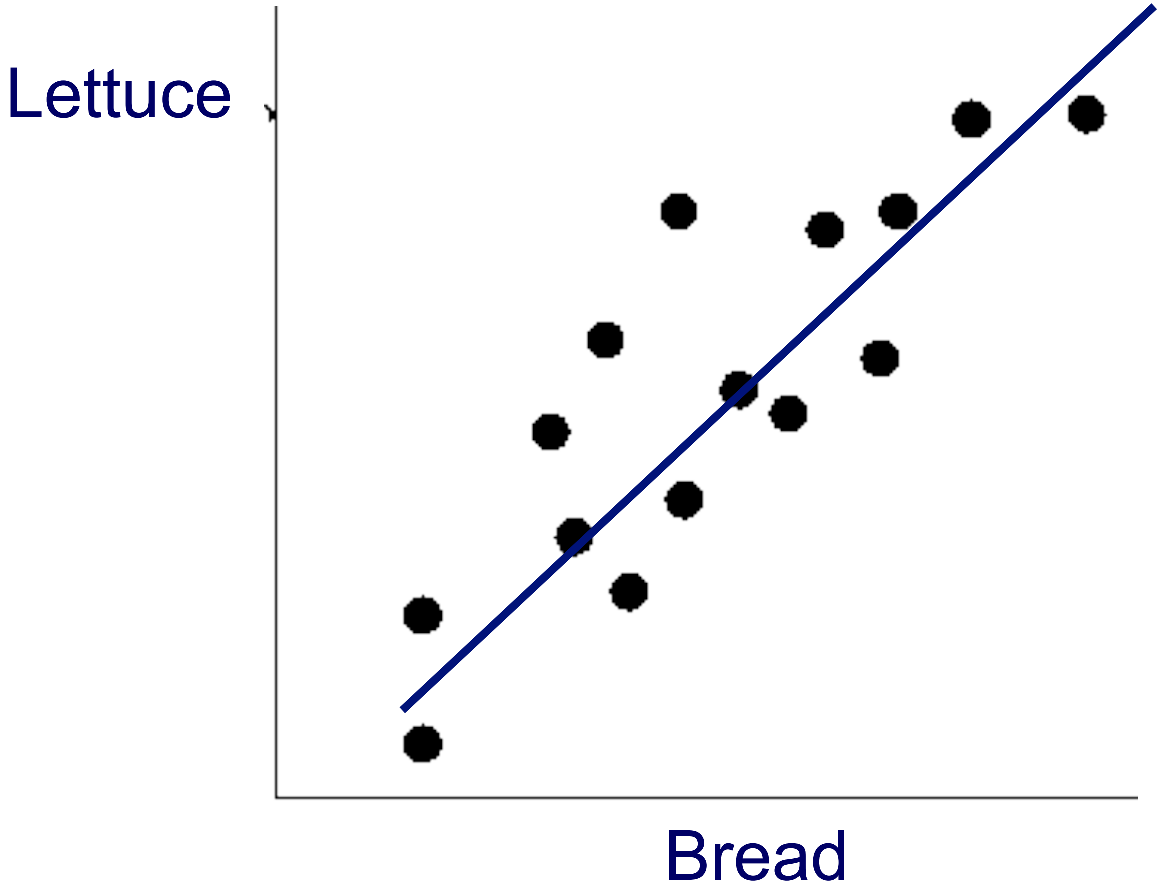
$$\begin{array}{c|ccccc} & \text{bread} & \text{lettuce} & \text{tomatos} & \text{beef} & \text{chicken} \\ \hline \text{vegetarians} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{meat eaters} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array}$$



SVD - Motivation



SVD - Motivation



details

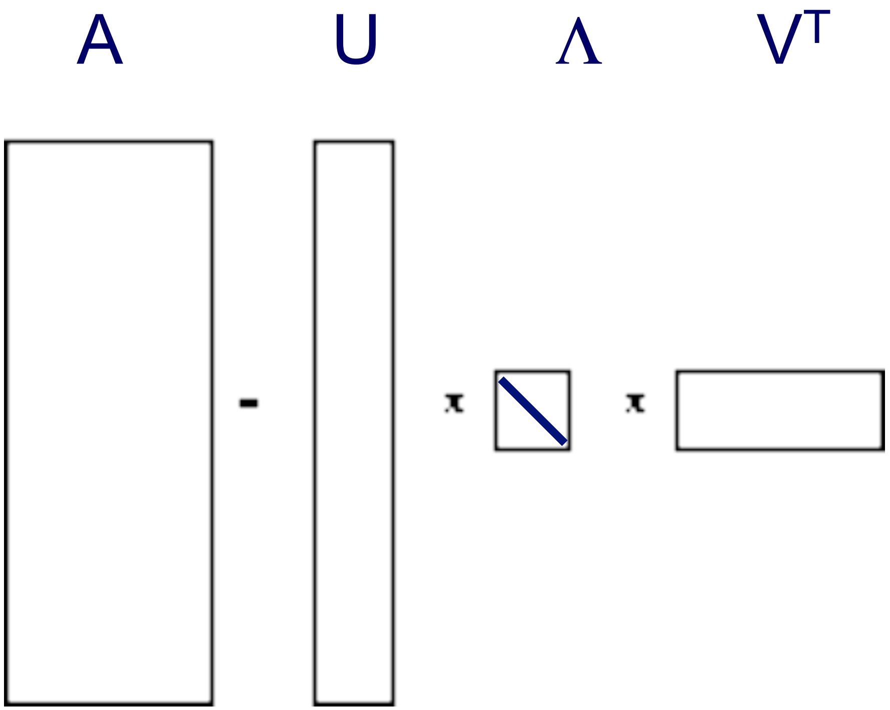
SVD - Definition

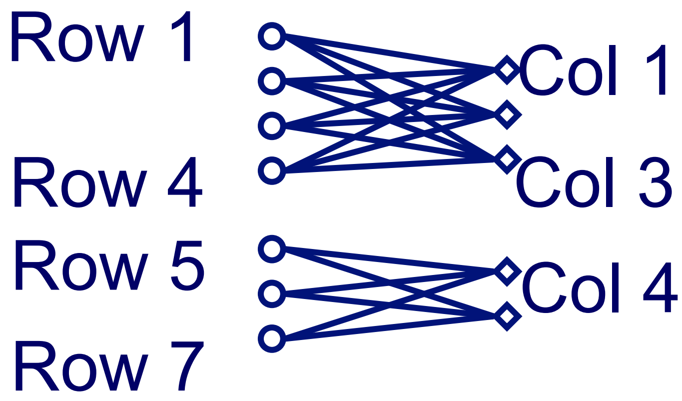
$$A_{[n \times m]} = U_{[n \times r]} \, \Lambda_{[r \times r]} \, (V_{[m \times r]})^T$$

- **A**: n x m matrix (eg., n customers, m products)
- **U**: n x r matrix (n customers, r concepts)
- **Λ**: r x r diagonal matrix (strength of each ‘concept’) (r : rank of the matrix)
- **V**: m x r matrix (m products, r concepts)

SVD - Definition

- $\mathbf{A} = \mathbf{U} \, \mathbf{\Lambda} \, \mathbf{V}^T$ - example:





details

SVD - Properties

THEOREM [Press+92]: always possible to decompose matrix **A** into **A** = **U** **Λ** **V**^{T} , where

- **U, Λ, V**: unique (*)
- **U**, **V**: column orthonormal (ie., columns are unit vectors, orthogonal to each other)
  - **U**^{T} **U** = **I**; **V**^{T} **V** = **I** (**I:** identity matrix**)**
- **Λ:** singular are positive, and sorted in decreasing order

SVD - Example

- $\mathbf{A} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^T$ - example:

$$\begin{array}{c|ccccc} & \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\ \hline \text{Veg.} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{Carn.} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Example

- $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$ - example:

Customer-to-Concept matrix

$$\begin{array}{cccccc} & \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\ \text{Veg.} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{Carn.} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Example

- **A** = **U Λ V**^{T} - example:

$$
\begin{array}{r|ccccc}
 & \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\
\hline
\text{Veg.} & 1 & 1 & 1 & 0 & 0 \\
 & 2 & 2 & 2 & 0 & 0 \\
 & 1 & 1 & 1 & 0 & 0 \\
 & 5 & 5 & 5 & 0 & 0 \\
\text{Carn.} & 0 & 0 & 0 & 2 & 2 \\
 & 0 & 0 & 0 & 3 & 3 \\
 & 0 & 0 & 0 & 1 & 1
\end{array}
=
\overset{\text{veg. concept}}{\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

- $\mathbf{A} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^{T}$ - example:

carn. concept

$$\begin{array}{c} \\ \text{Veg.} \\ \\ \\ \\ \text{Carn.} \\ \\ \\ \end{array}\left[\begin{array}{ccccc} \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\ 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right] = \left[\begin{array}{cc} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{array}\right] \times \left[\begin{array}{cc} 9.64 & 0 \\ 0 & 5.29 \end{array}\right] \times \left[\begin{array}{ccccc} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{array}\right]$$

SVD - Example

- $\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{V}^T$ - example:

strength of concept

$$\begin{array}{c|ccccc} & \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\ \hline \text{Veg.} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{Carn.} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} \boxed{9.64} & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

SVD - Example

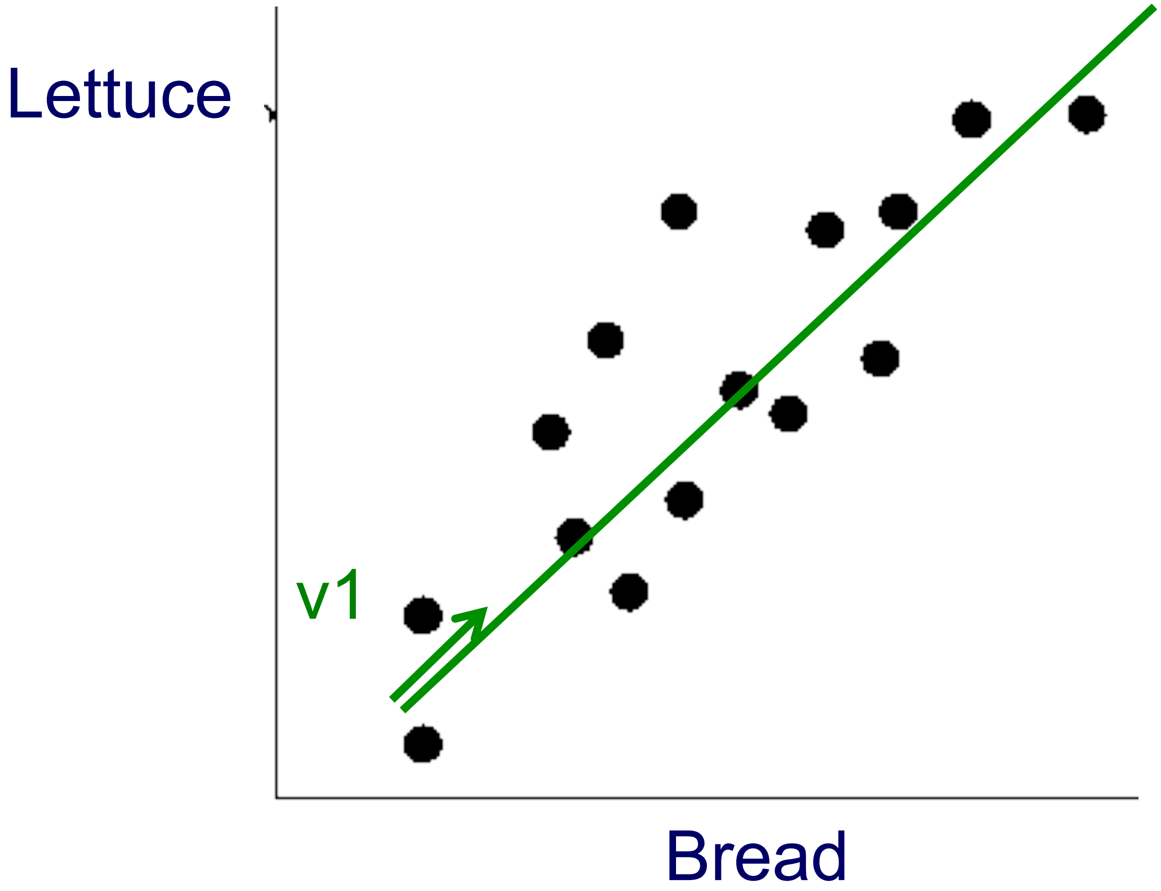
- $\mathbf{A} = \mathbf{U}\, \mathbf{\Lambda}\, \mathbf{V}^T$ - example:

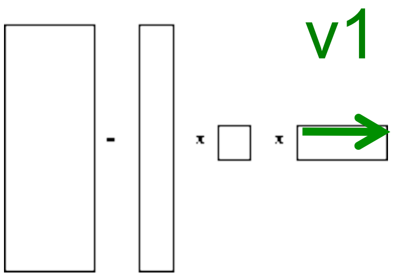
prod.-to-concept
similarity matrix

$$\begin{array}{c|ccccc} & \text{bread} & \text{lett.} & \text{tom.} & \text{beef} & \text{chick.} \\ \hline \text{Veg.} & 1 & 1 & 1 & 0 & 0 \\ & 2 & 2 & 2 & 0 & 0 \\ & 1 & 1 & 1 & 0 & 0 \\ & 5 & 5 & 5 & 0 & 0 \\ \text{Carn.} & 0 & 0 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 3 & 3 \\ & 0 & 0 & 0 & 1 & 1 \end{array} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

v1

SVD - Motivation



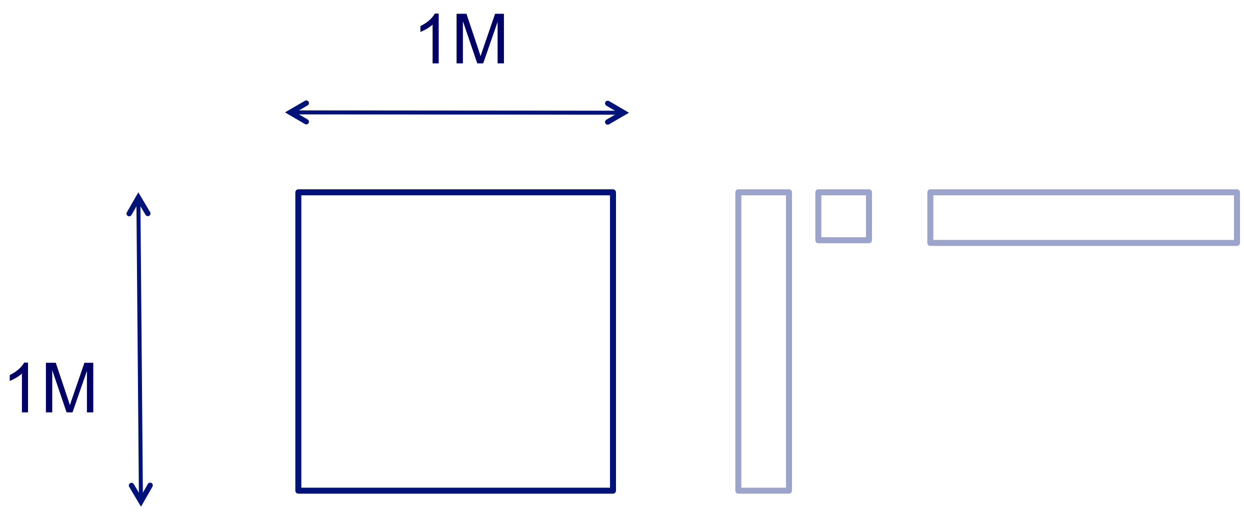


EigenSpokes

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.

Motivation

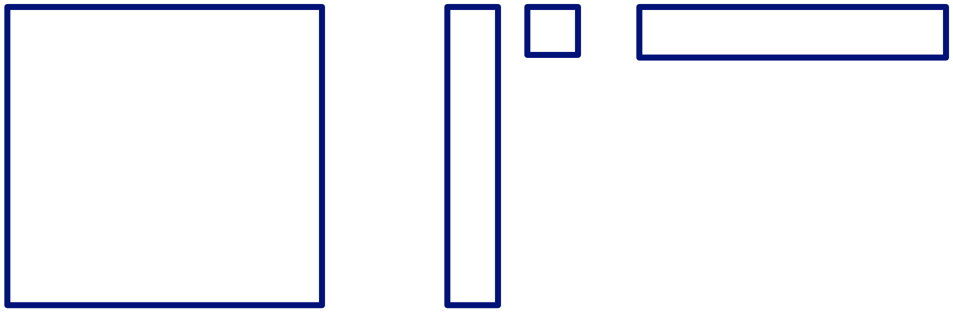
- Who-calls-whom (~1M) – how to visualize/ understand?



EigenSpokes

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

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

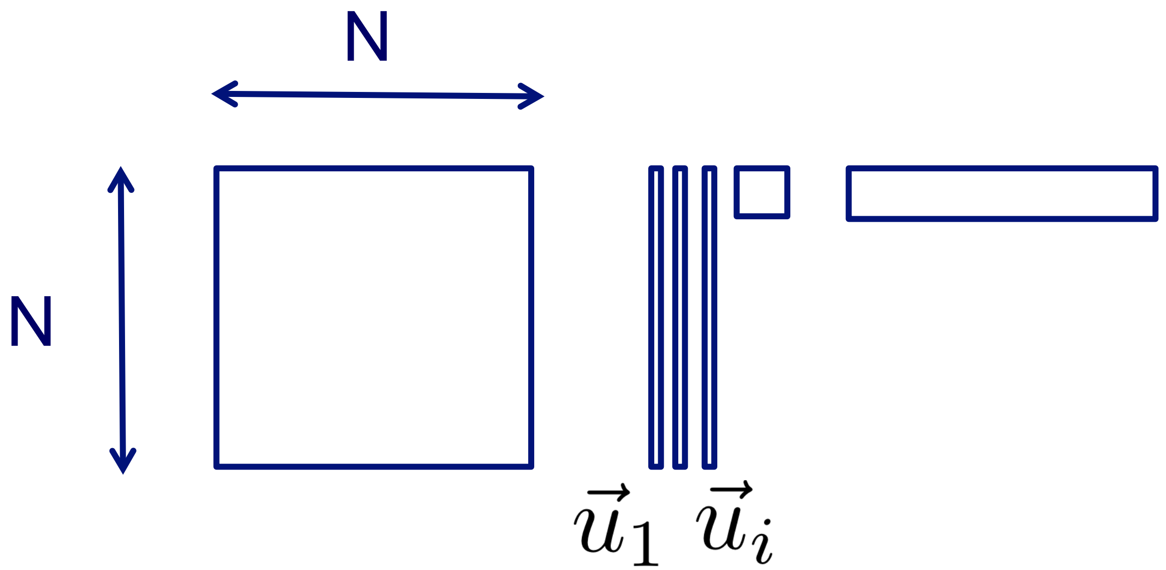


details

EigenSpokes

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

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

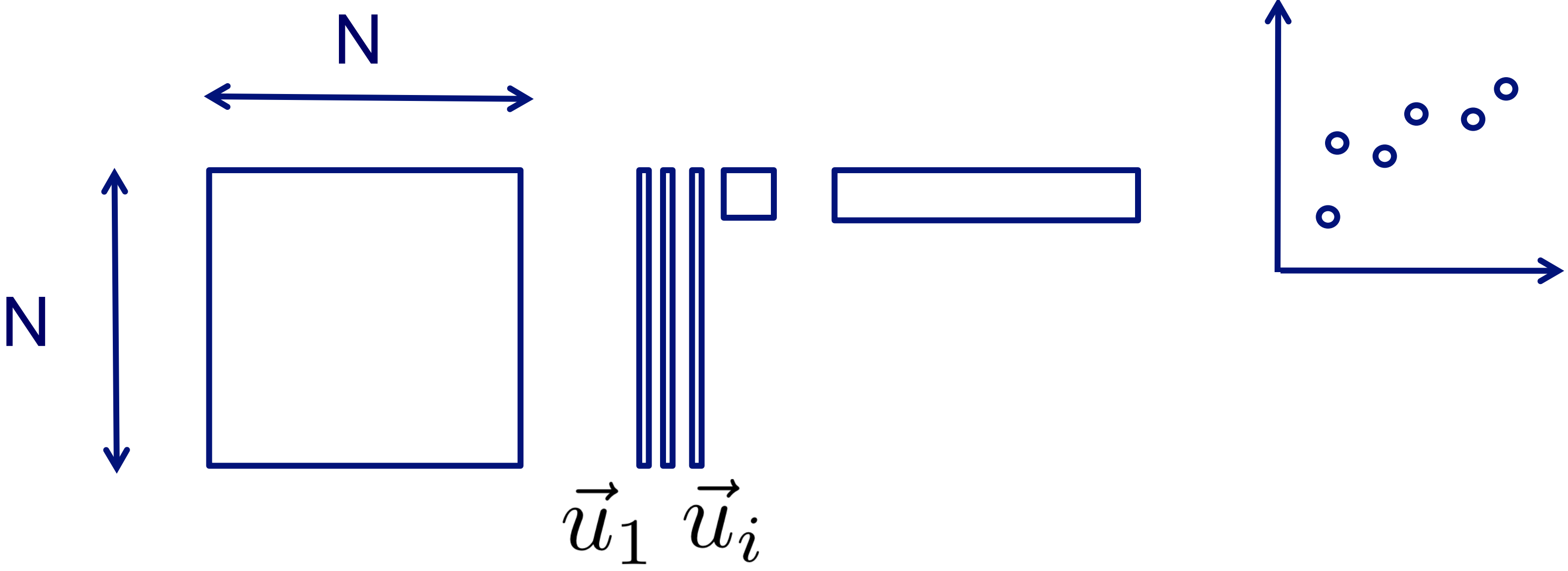


details

EigenSpokes

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

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

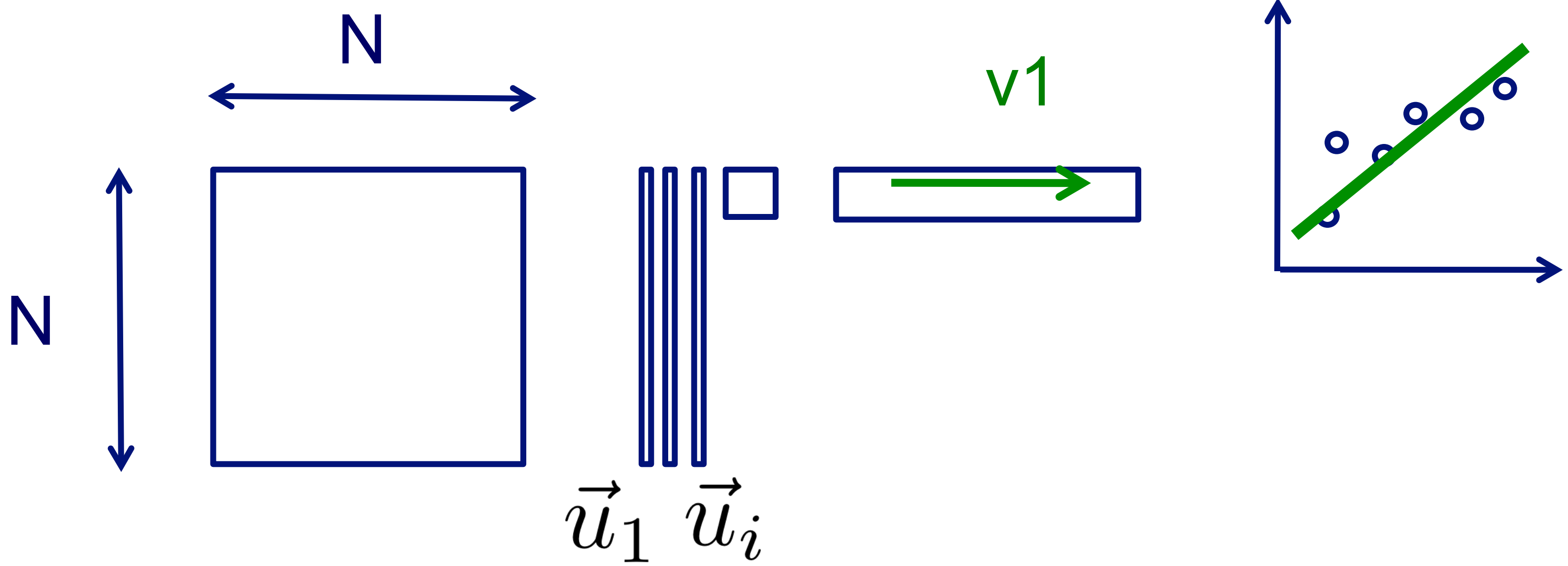


details

EigenSpokes

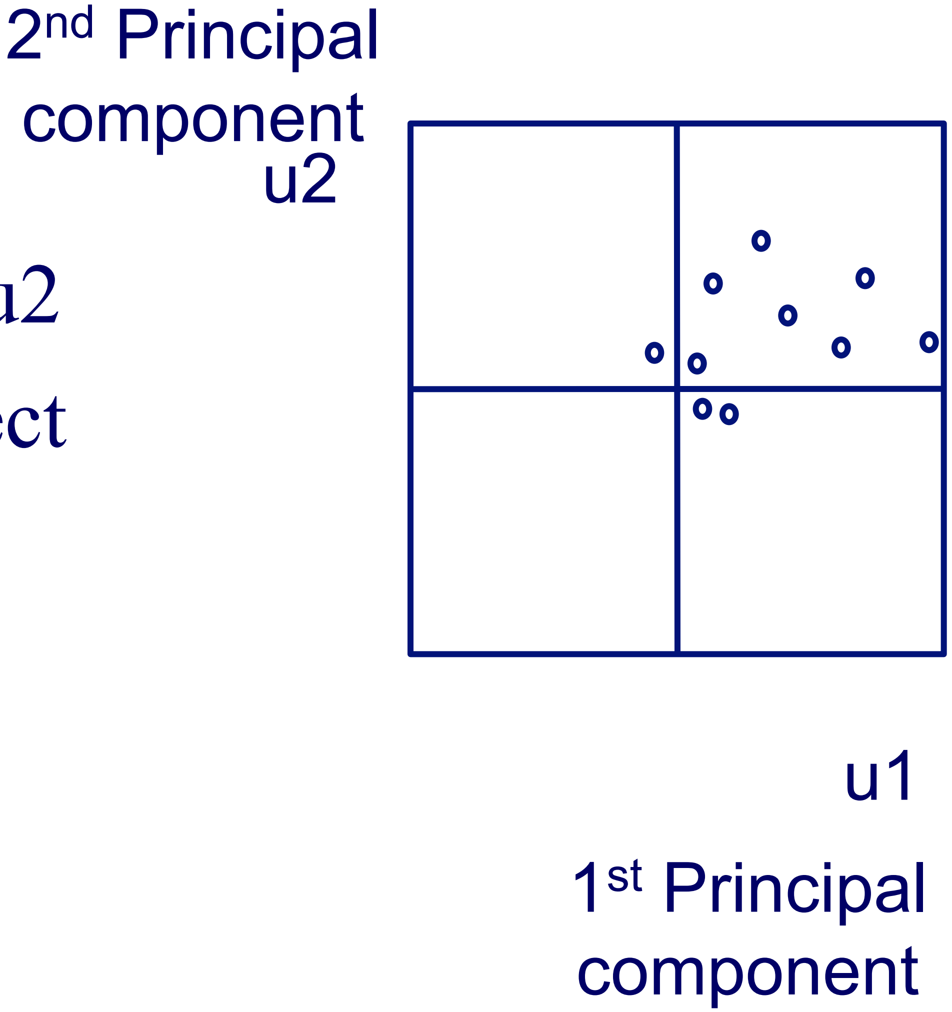
- Eigenvectors of adjacency matrix
    - equivalent to singular vectors (symmetric, undirected graph)

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



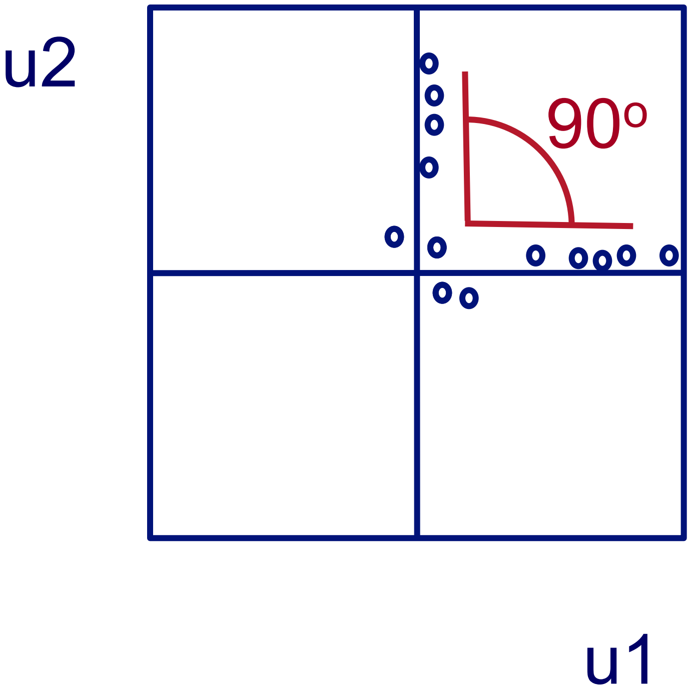
EigenSpokes

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



EigenSpokes

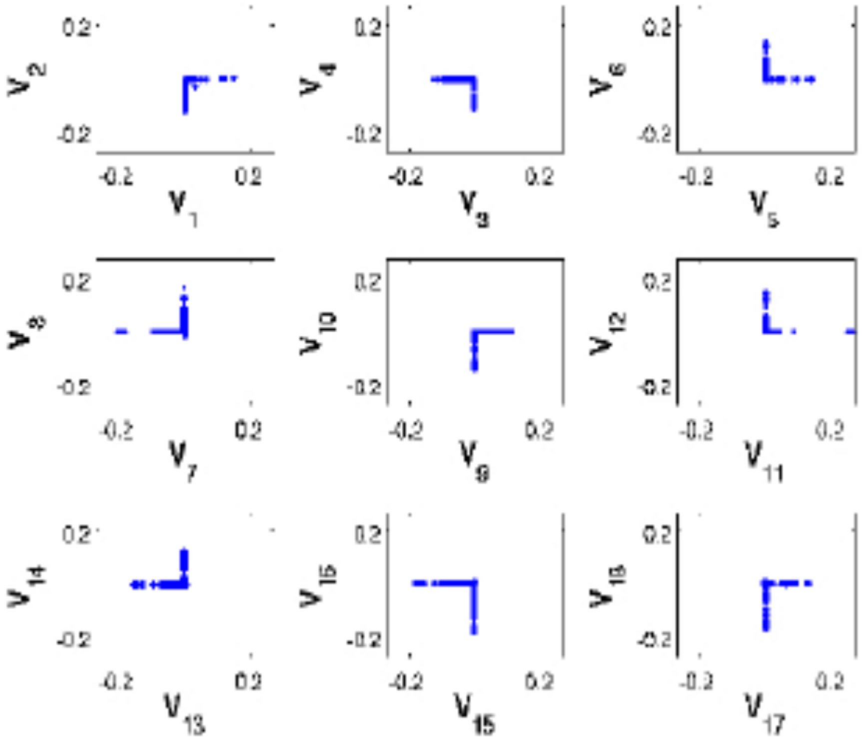
- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A f[illegible]ttered ~r[illegible]mly



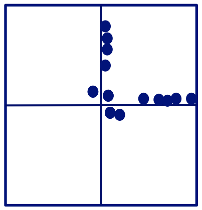
Carnegie Mellon

EigenSpokes - pervasiveness

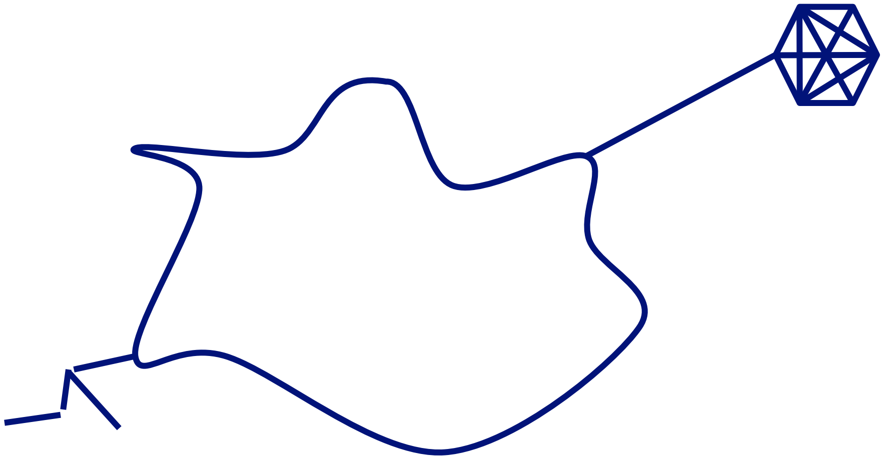
- Present in mobile social graph
  - across time and space
- Patent citation graph



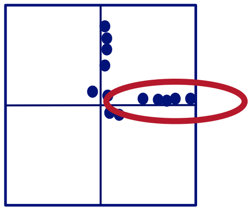
EigenSpokes - explanation



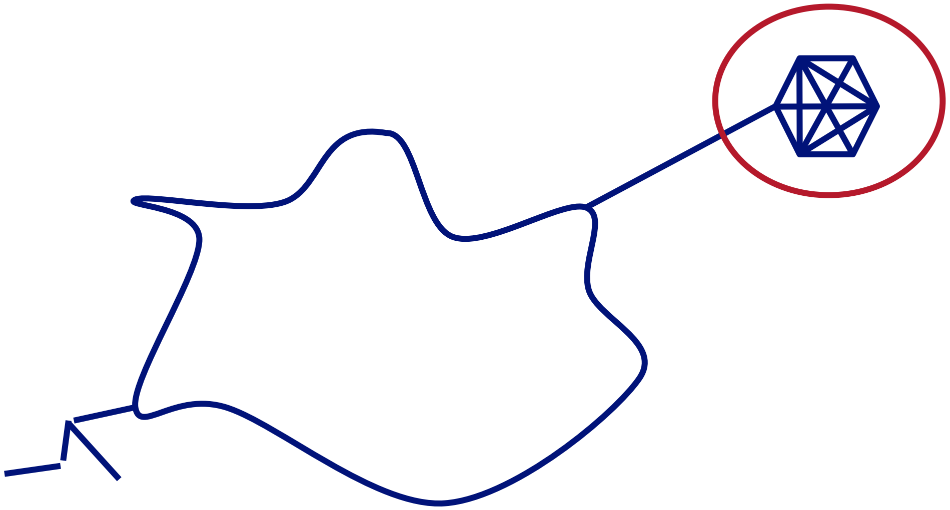
Near-cliques, or near-bipartite-cores, loosely connected



EigenSpokes - explanation

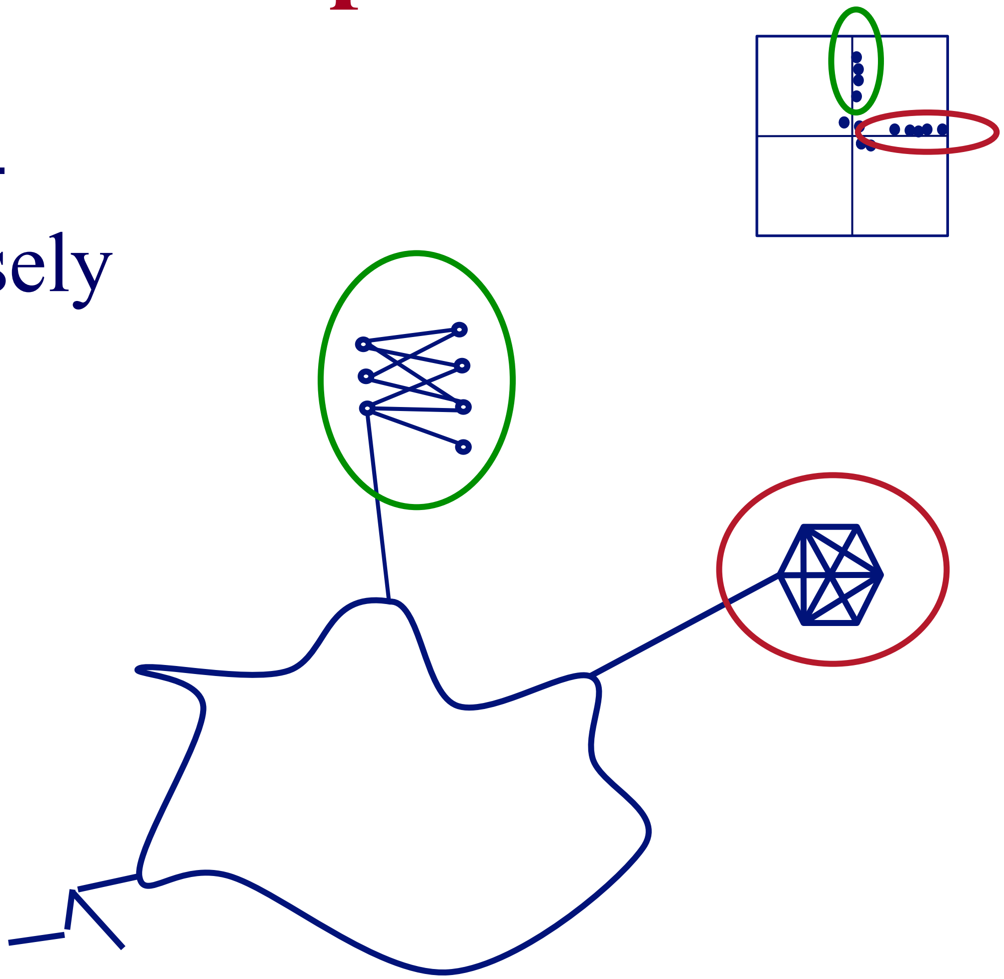


Near-cliques, or near-bipartite-cores, loosely connected



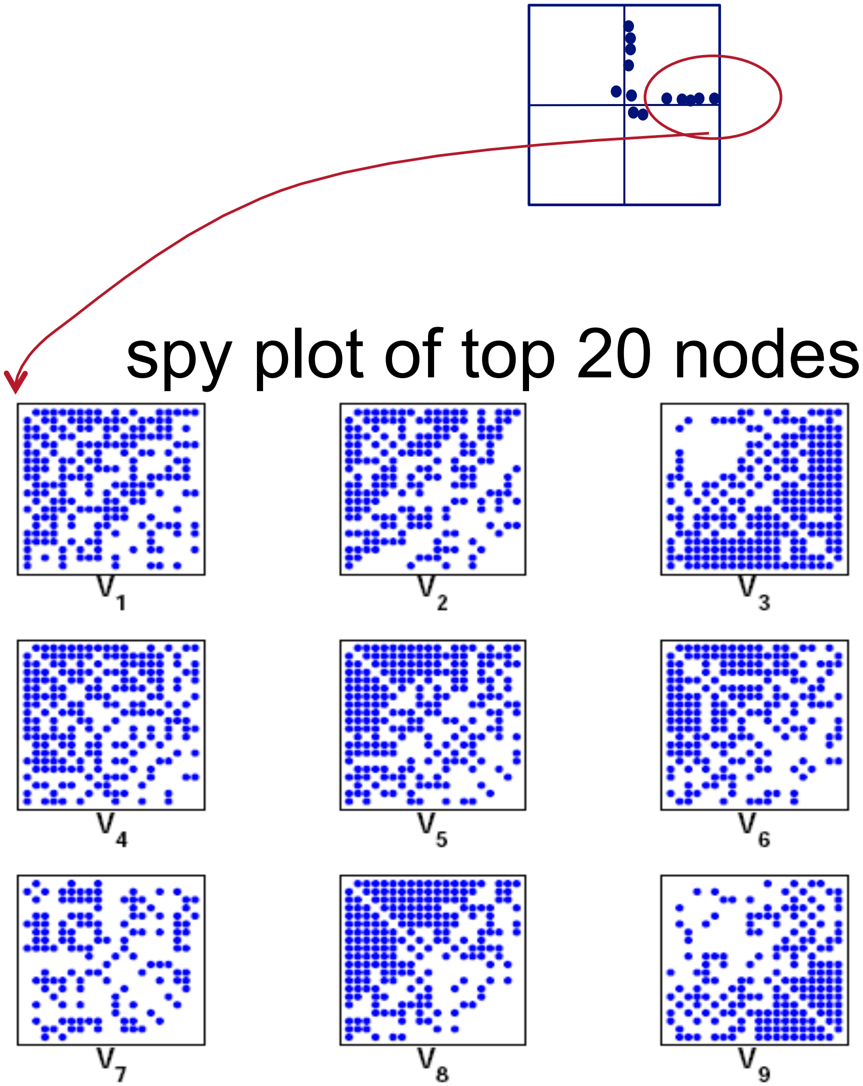
EigenSpokes - explanation

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



EigenSpokes - explanation

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


spy plot of top 20 nodes

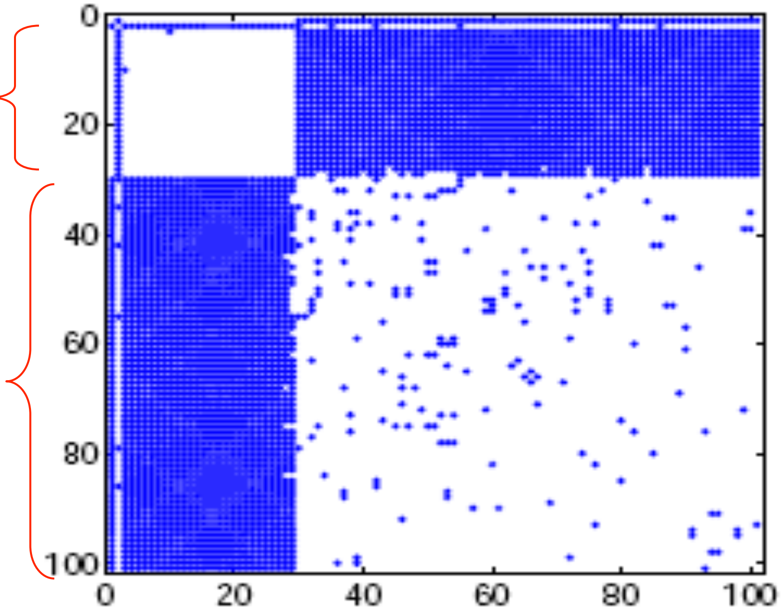
So what?

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

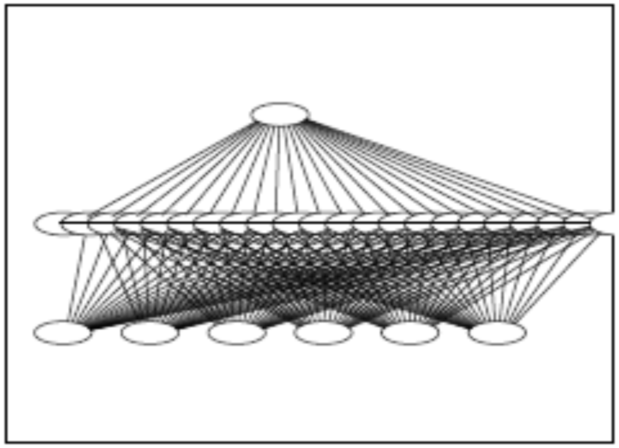
Bipartite Communities!

patents from
same inventor(s)

cut-and-paste
bibliography!

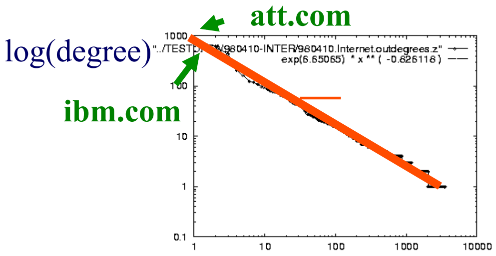


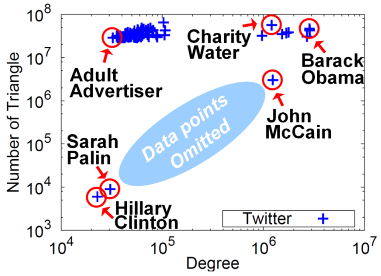
magnified bipartite community



Outline

- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Oddball
  - SVD
- ➡ Problem#3: Algorithms and Scalability
- Conclusions





PROJECT PEGASUS

www.cs.cmu.edu/~pegasus

QUESTIONS ?