Carnegie Mellon

Mining Large Graphs and Tensors - Patterns, Tools and Discoveries.

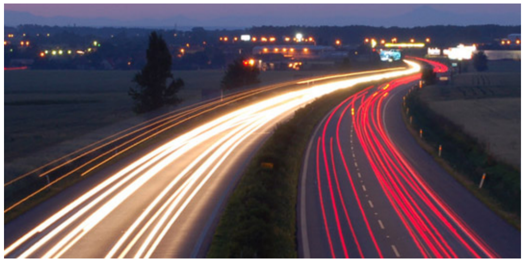
Christos Faloutsos

CMU

Thank you!

- Nikos Sidiropoulos
- Kuo-Chu Chang
- Zhi (Gerry) Tian

Roadmap



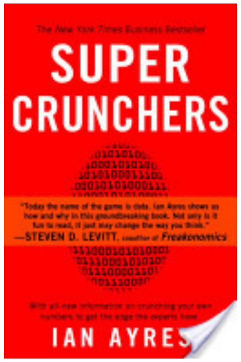
- ➡ Introduction – Motivation
  - Why ‘big data’
  - Why (big) graphs?
- Problem#1: Patterns in graphs
- Problem#2: Tools
- Conclusions

Why 'big data'

- Why?
- What is the problem definition?

Main message:
Big data: often > experts

- ‘Super Crunchers’ *Why Thinking-By-Numbers is the New Way To Be Smart* by Ian Ayres, 2008

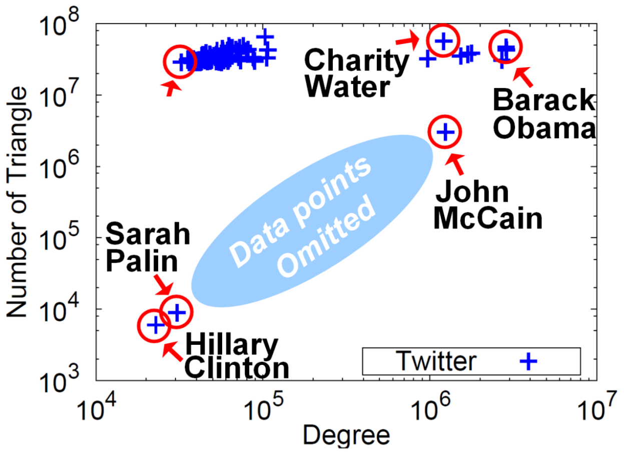


- Google won the machine translation competition 2005
- http://www.itl.nist.gov/iad/mig//tests/mt/2005/doc/mt05eval_official_results_release_20050801_v3.html

Problem definition – big picture

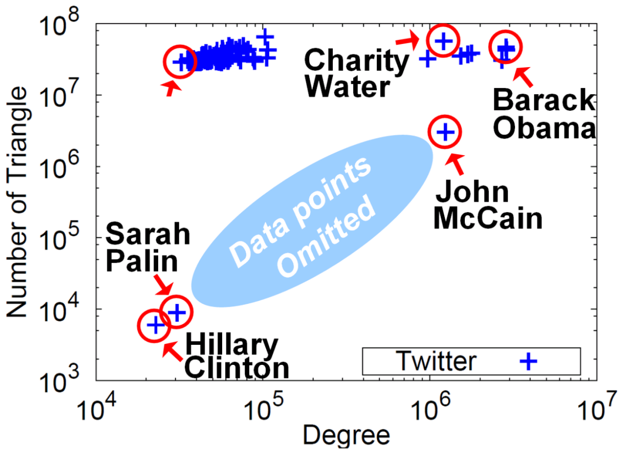
Tera/Peta-byte data

Analytics



Insights, outliers

Problem definition – big picture



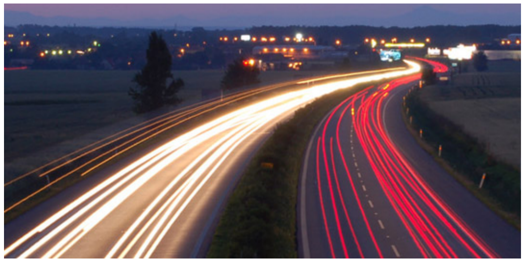
Tera/Peta-byte data

Analytics

Insights, outliers

Main emphasis in this talk

Roadmap

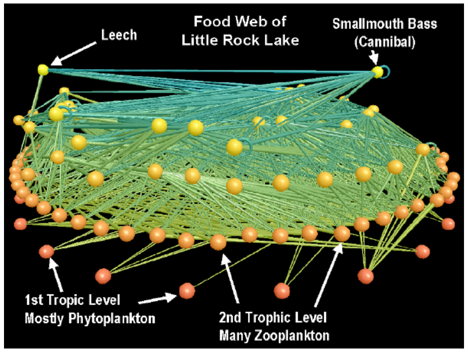


- Introduction – Motivation
  - Why 'big data'
  - ➡ Why (big) graphs?
- Problem#1: Patterns in graphs
- Problem#2: Tools
- Problem#3: Scalability
- Conclusions

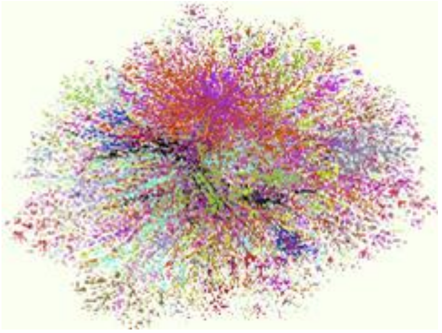
Graphs - why should we care?

>$10B revenue

>0.5B users



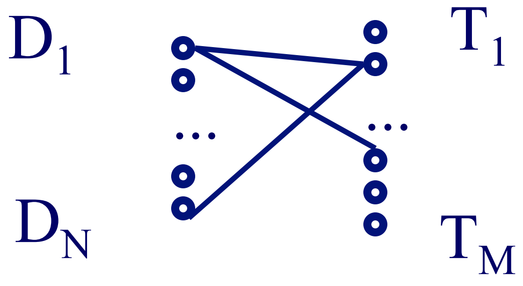
Food Web
[Martinez '91]



Internet Map
[lumeta.com]

Graphs - why should we care?

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



- web: hyper-text graph

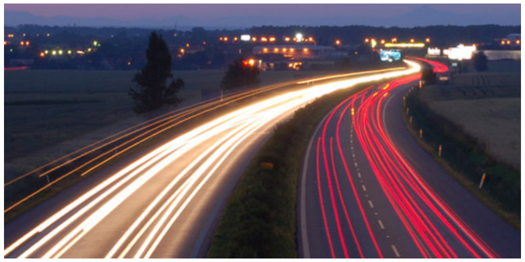
- ... and more:

Graphs - why should we care?

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

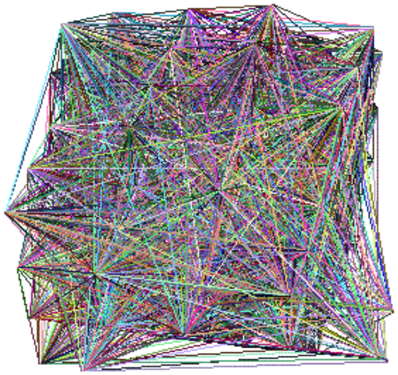
Carnegie Mellon

Outline



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

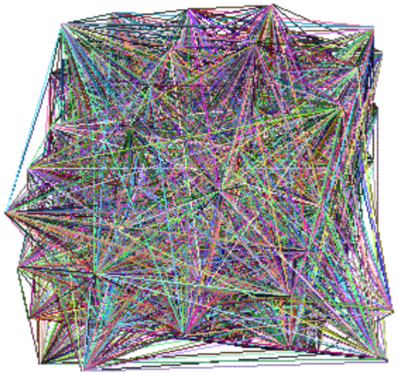
Problem #1 - network and graph mining



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

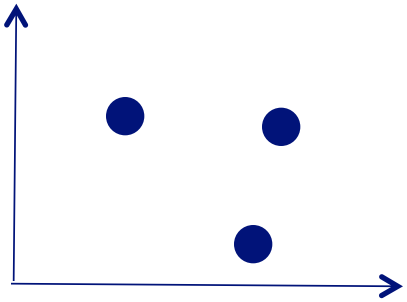
- What is ‘normal’/‘abnormal’?
- which patterns/laws hold?

Problem #1 - network and graph mining

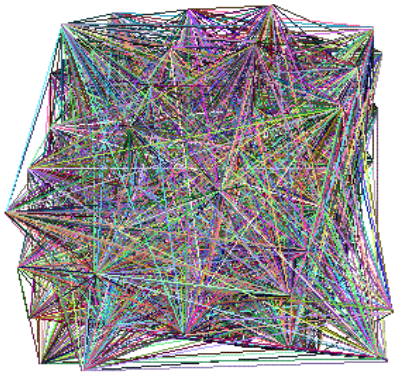


- What does the Internet look like?
- What does FaceBook look like?

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

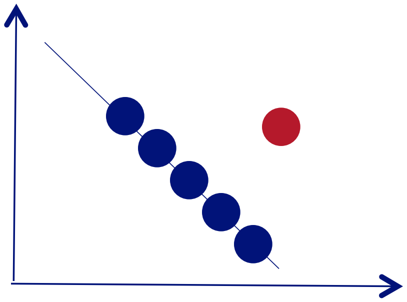


Problem #1 - network and graph mining



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

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



Graph mining

- Are real graphs random?

Laws and patterns

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

- So, let’s look at the data

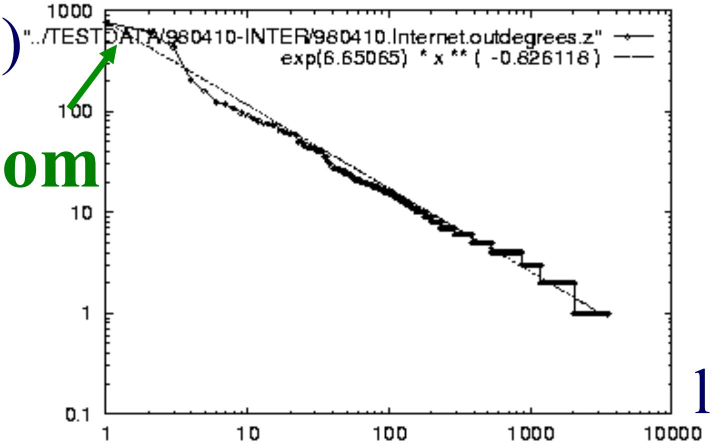
Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

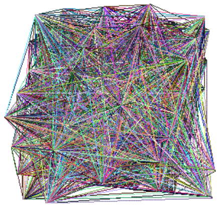
internet domains

att.com

log(degree)



ibm.com

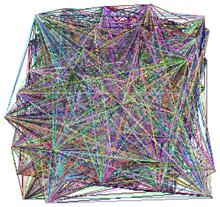


log(rank)

Solution# S.1

- Power law in the degree distribution [SIGCOMM99]

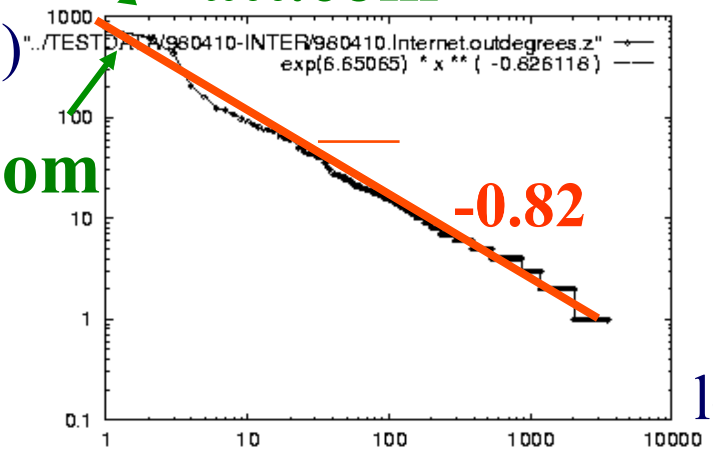
internet domains



att.com

log(degree)

ibm.com



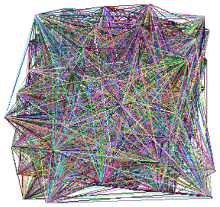
-0.82

log(rank)

Solution# S.1

- Q: So what?

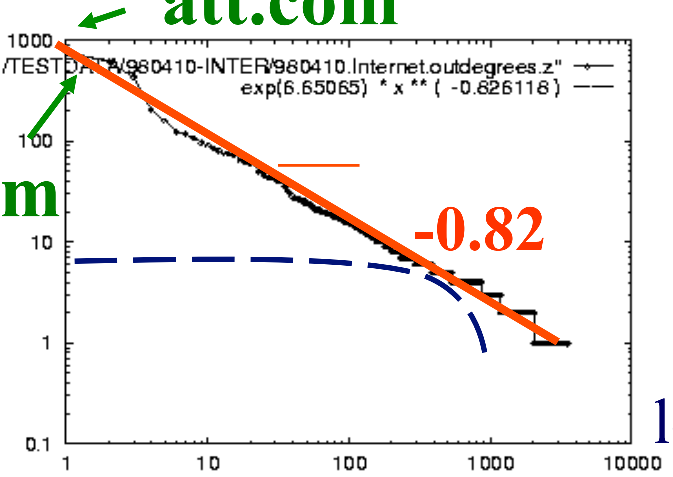
internet domains



att.com

log(degree)

ibm.com



-0.82

log(rank)

Carnegie Mellon

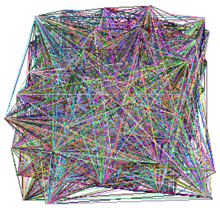
Solution# S.1

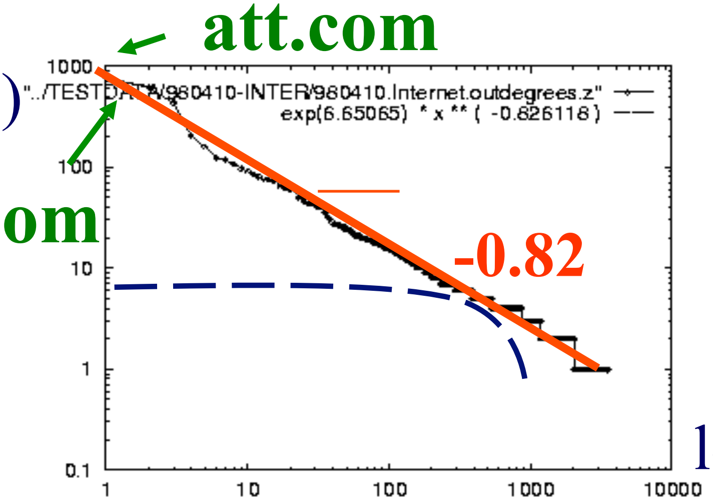
- Q: So what?
- A1: # of two-step-away pairs: O(d_max ^2) ~ 10M^2

⇩

~0.8PB ->
a data center(!)

internet domains




att.com
log(degree)
ibm.com
"../TESTDATA/980410-INTER/980410.Internet.outdegrees.z"
exp(6.65065) * x ** (-0.826118)
-0.82
1000
100
10
1
0.1
1
10
100
1000
10000
log(rank)

Solution# S.1

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

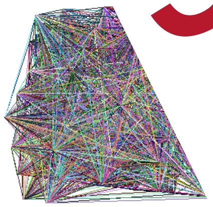
inte

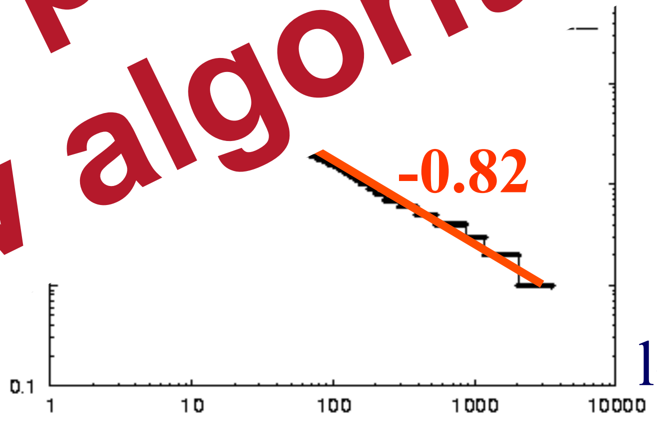
Such patterns ->

New algorithms

⇩

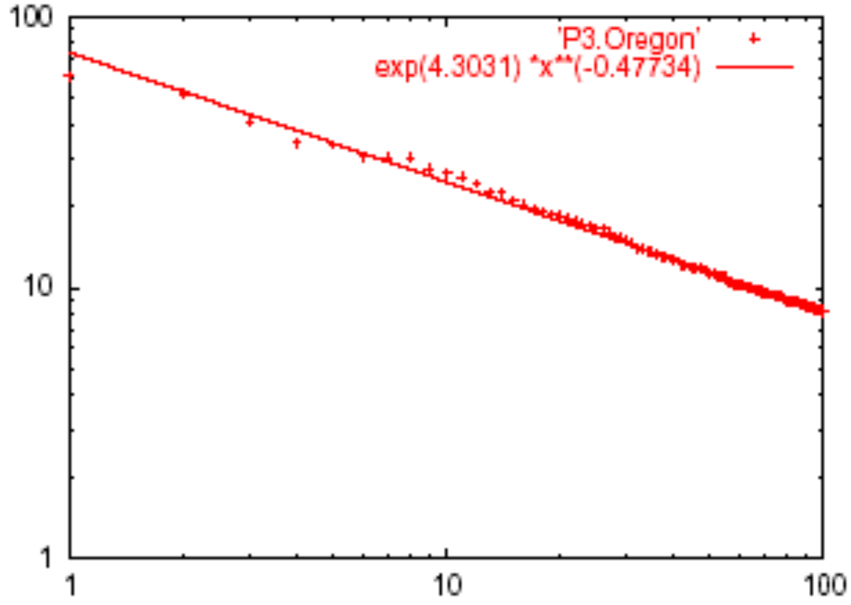
~0.8PB ->
a data center(!)





Solution# S.2: Eigen Exponent *E*

Eigenvalue



Exponent = slope

E = -0.48

May 2001

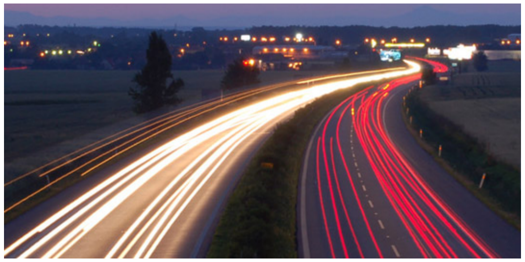
Rank of decreasing eigenvalue

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

Many more power laws

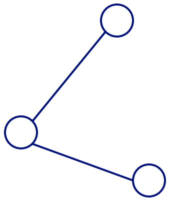
- # of sexual contacts
- Income [Pareto] –'80-20 distribution'
- Duration of downloads [Bestavros+]
- Duration of UNIX jobs ('mice and elephants')
- Size of files of a user
- …
- 'Black swans'

Roadmap



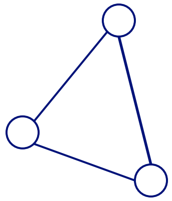
- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
    - degree, diameter, eigen,
    - ➡ triangles
    - cliques
  - Weighted graphs
  - Time evolving graphs
- Problem#2: Tools

Solution# S.3: Triangle ‘Laws’



- Real social networks have a lot of triangles

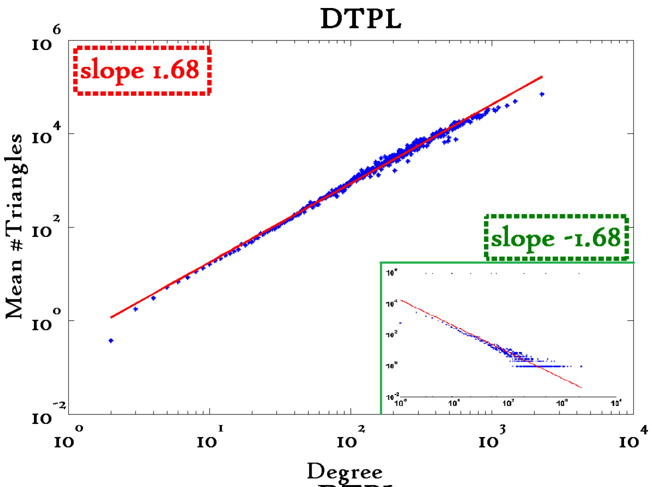
Solution# S.3: Triangle 'Laws'



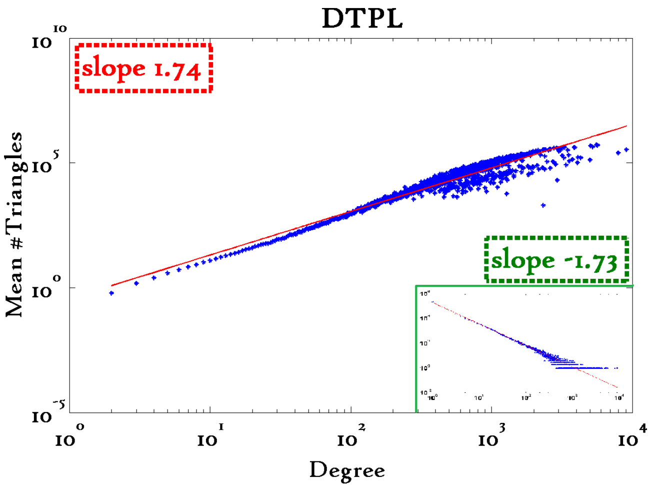
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?

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

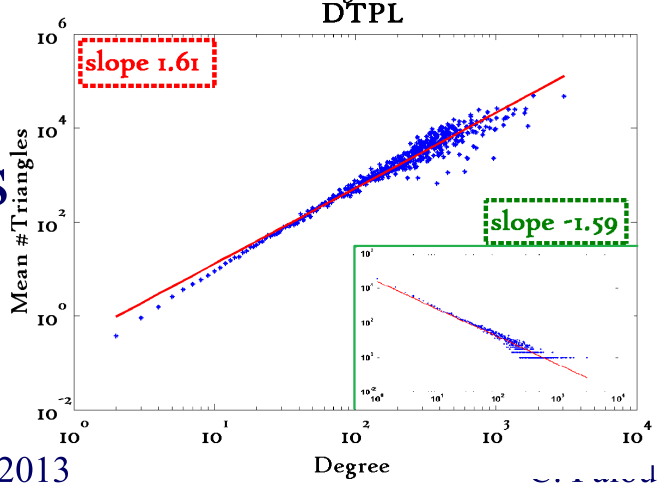
Reuters

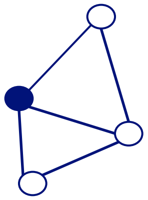


SN



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 (λ_i^3)

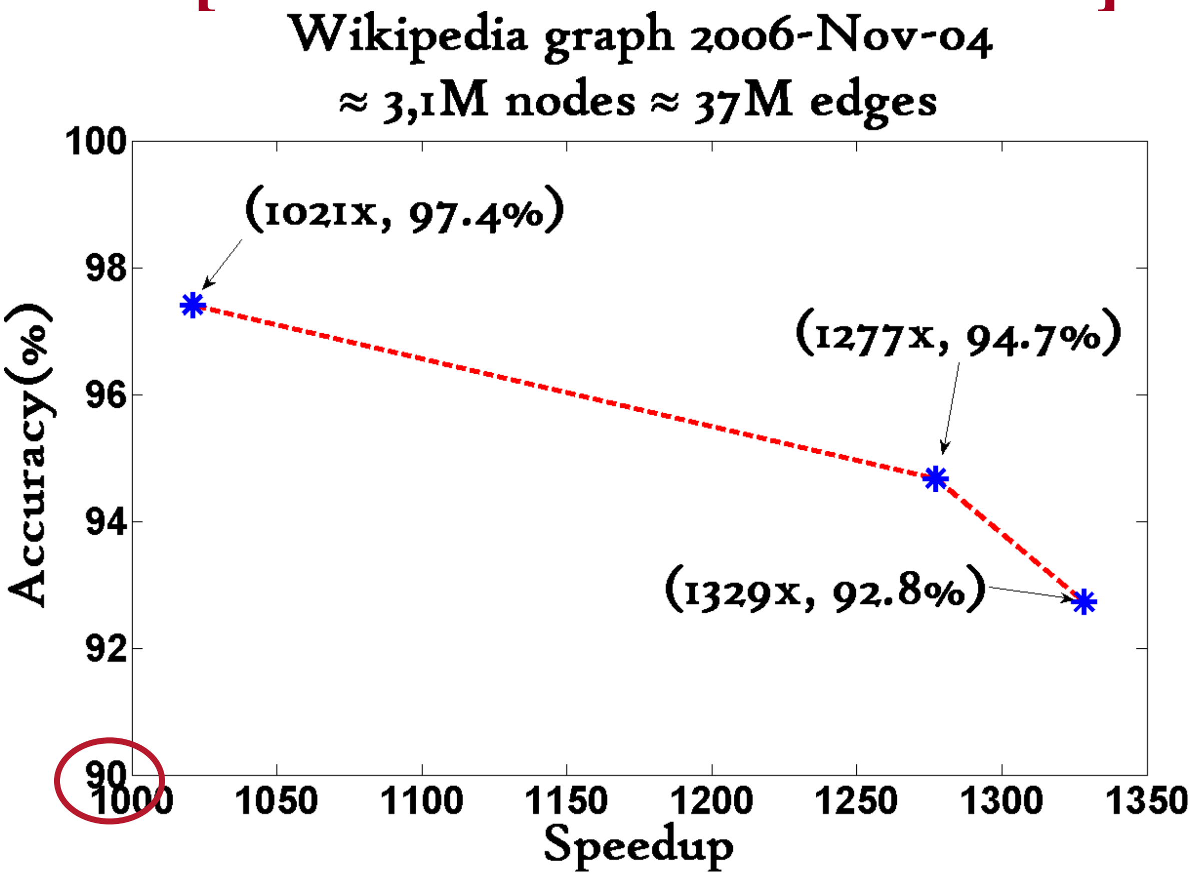
(and, because of skewness (S2) ,

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

Carnegie Mellon

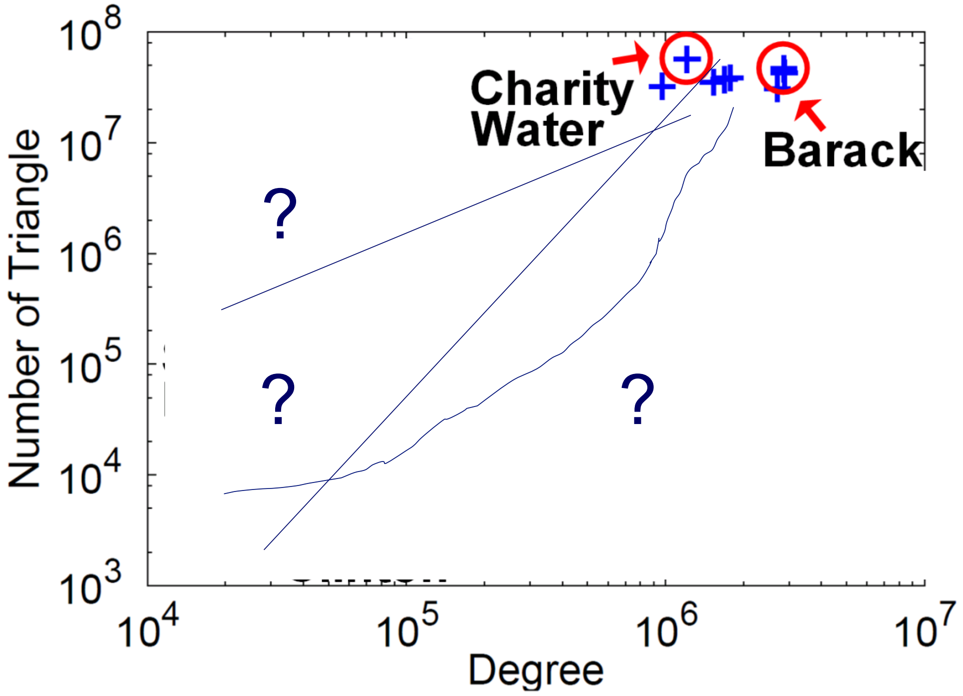
details

Triangle Law: Computations [Tsourakakis ICDM 2008]



1000x+ speed-up, >90% accuracy

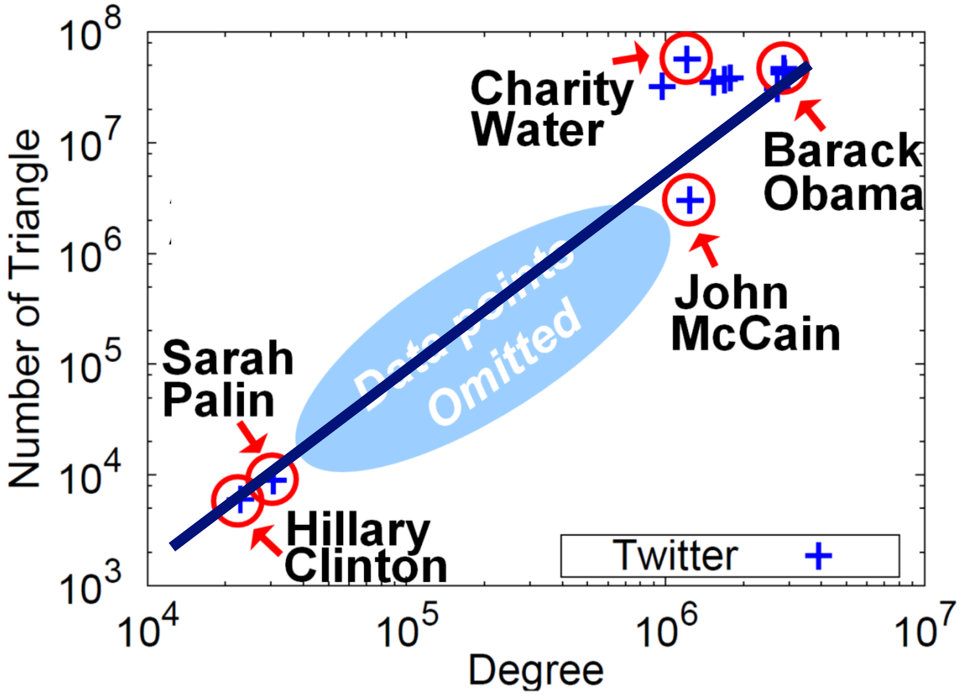
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

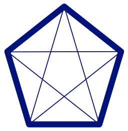
Triangle counting for large graphs?

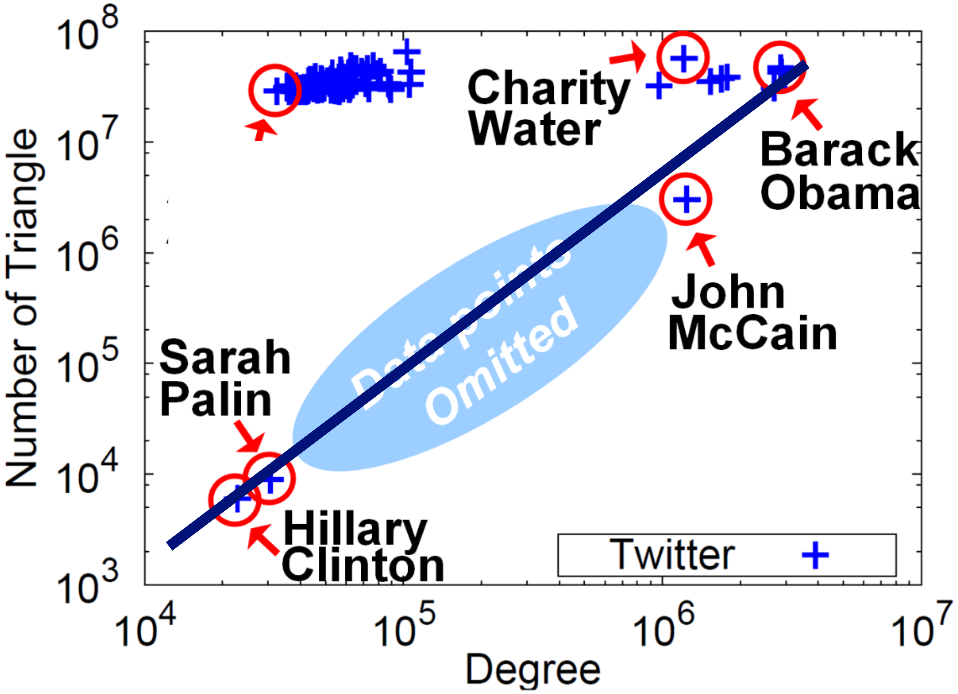


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?

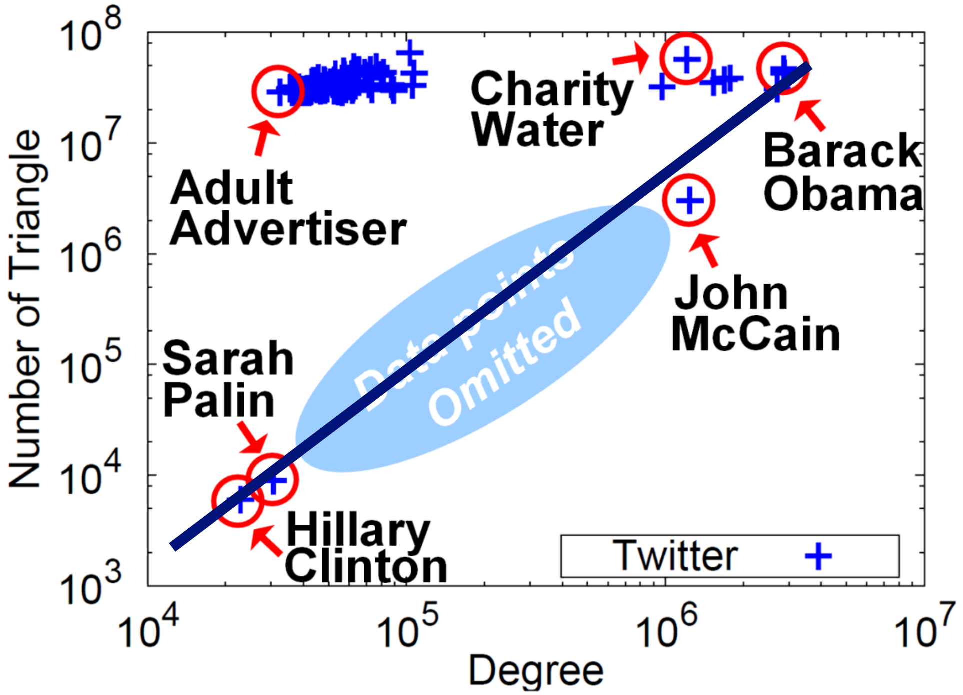




Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

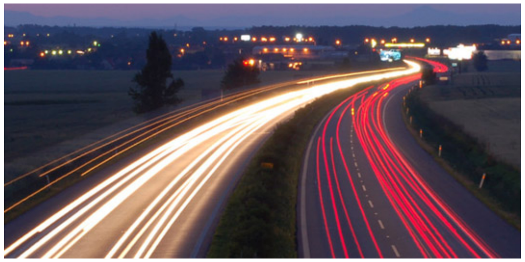
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

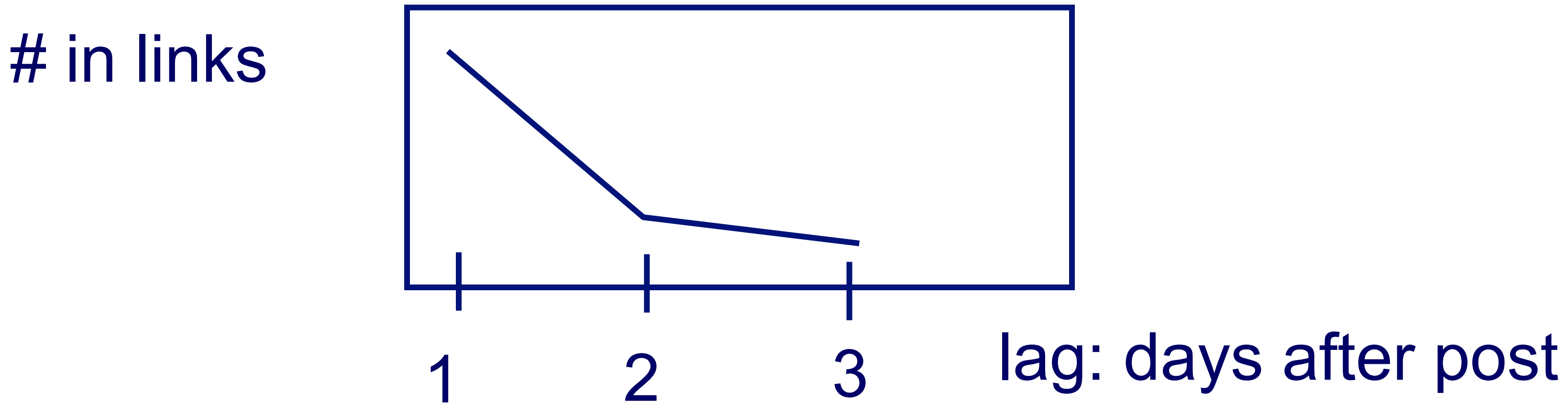
[U Kang, Brendan Meeder, +, PAKDD’11]

Roadmap

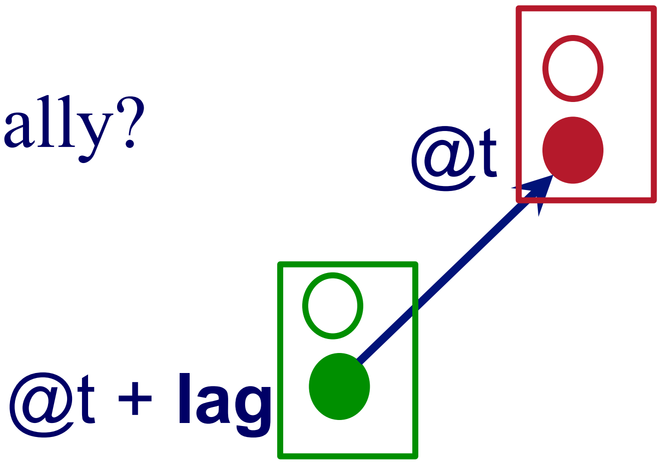


- Introduction – Motivation
- Problem#1: Patterns in graphs
  - Static graphs
  - ➡ Time evolving graphs
- Problem#2: Tools
- …

T.1 : popularity over time

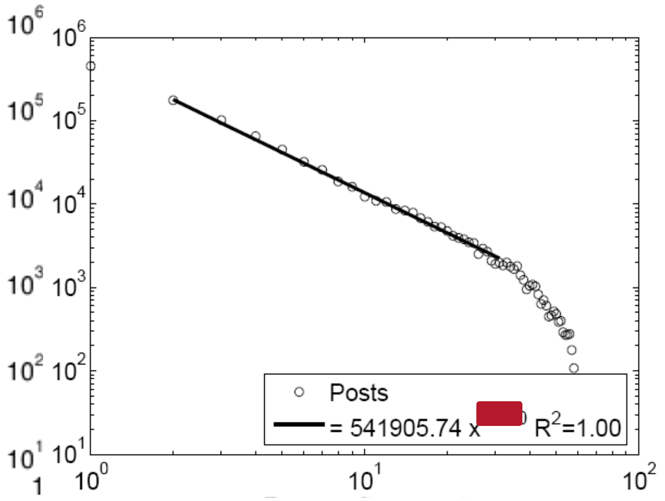


Post popularity drops-off – exponentially?



T.1 : popularity over time

# in links
(**log**)



days after post
(**log**)

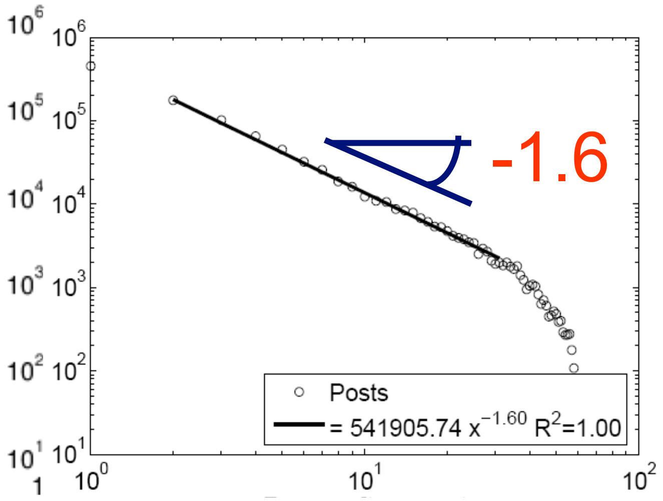
Post popularity drops-off – exponentially?

POWER LAW!

Exponent?

T.1 : popularity over time

# in links (**log**)



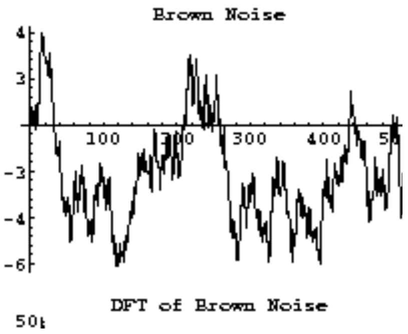
days after post (**log**)

Post popularity drops-off – exponentially? 🚫

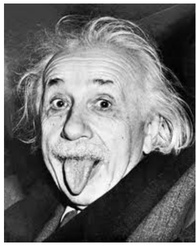
POWER LAW!

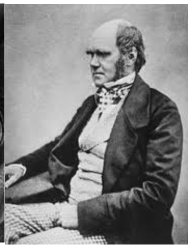
Exponent? -1.6

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

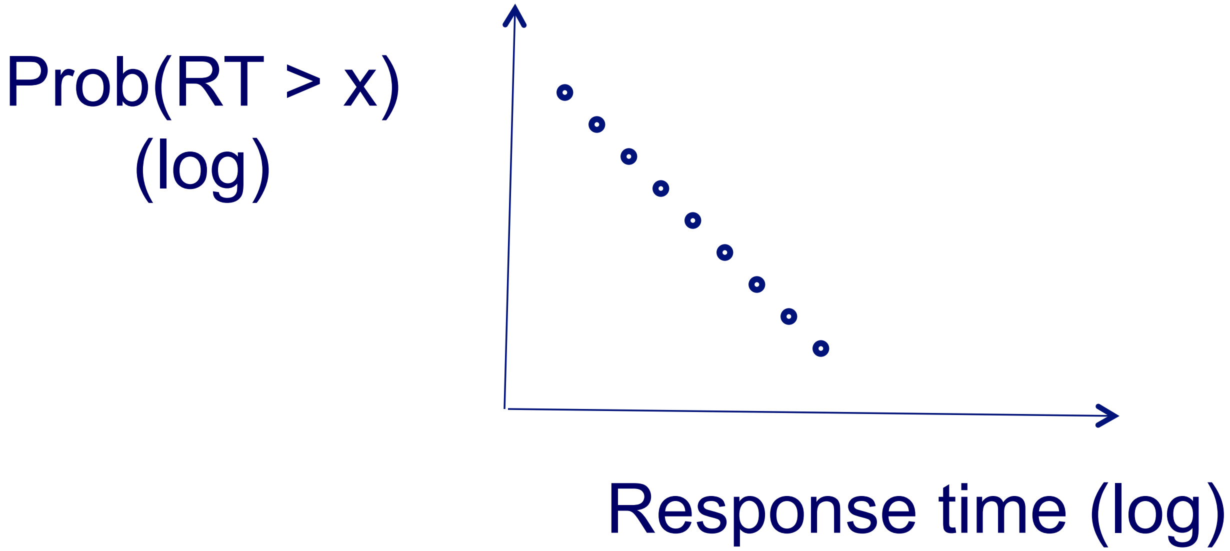


-1.5 slope

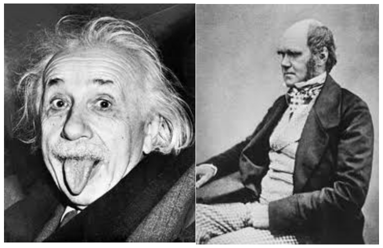




J. G. Oliveira & A.-L. Barabási Human Dynamics: The Correspondence Patterns of Darwin and Einstein. *Nature* **437,** 1251 (2005) . [PDF]

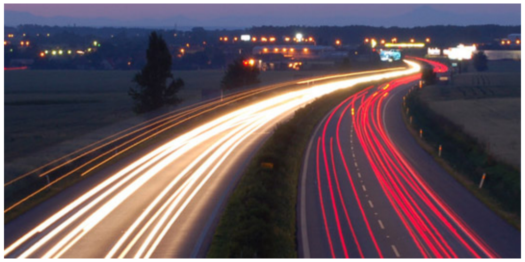


-1.5 slope



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

Roadmap



- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - (Belief Propagation)
  - ➡ Tensors
  - Spike analysis
- Conclusions

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

U Kang

Evangelos Papalexakis

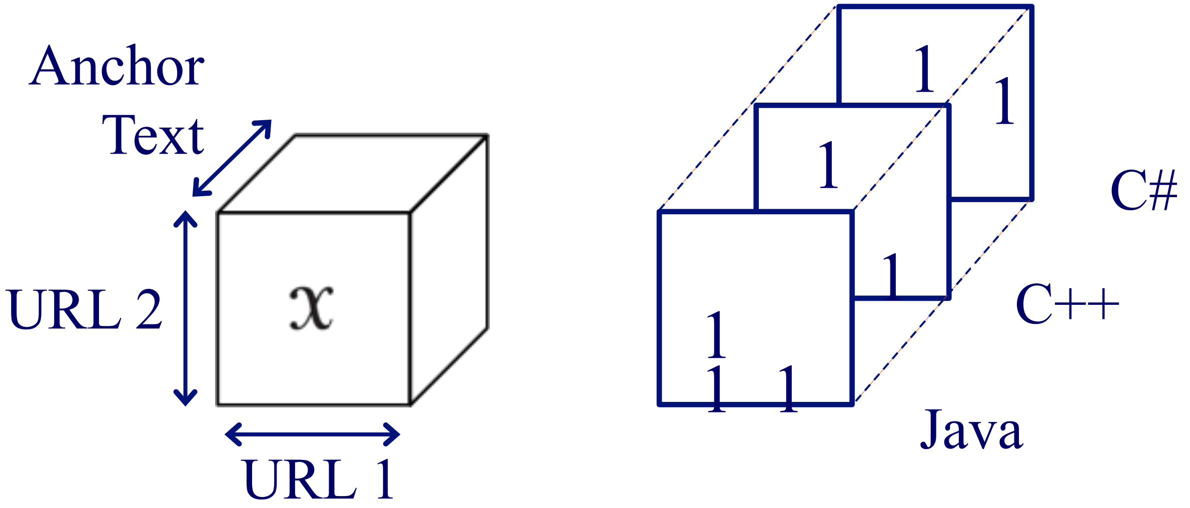
Abhay Harpale

Christos Faloutsos

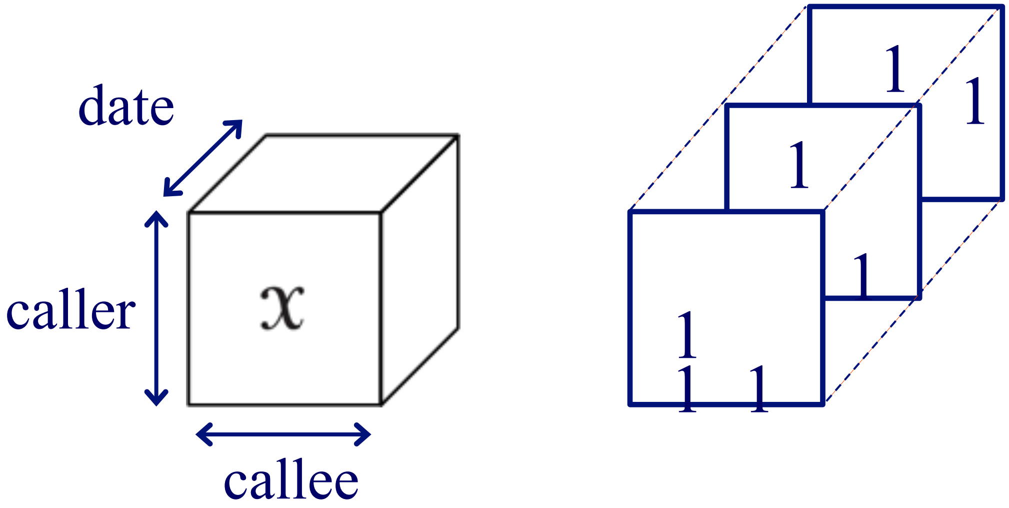
KDD'12

Background: Tensor

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



Time evolving graphs: Tensors

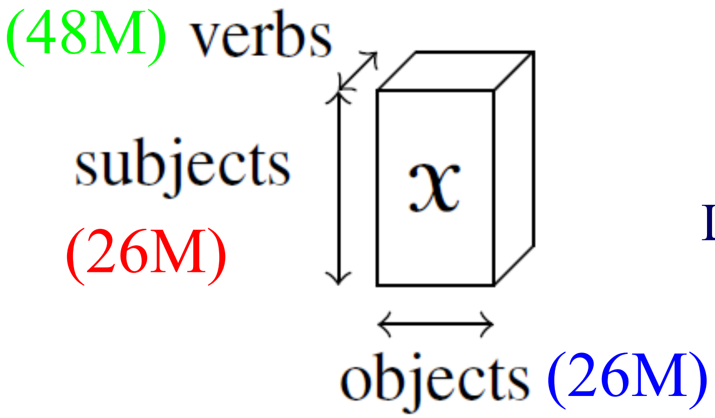


Background: Tensor

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

"Eric Clapton plays guitar"

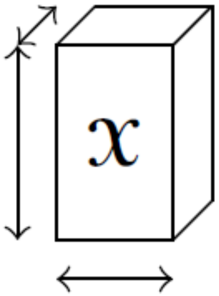
"Barrack Obama is the president of U.S."



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

Background: Tensor

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



Time-stamp

IP-source

IP-destination

Anomaly Detection in Computer networks

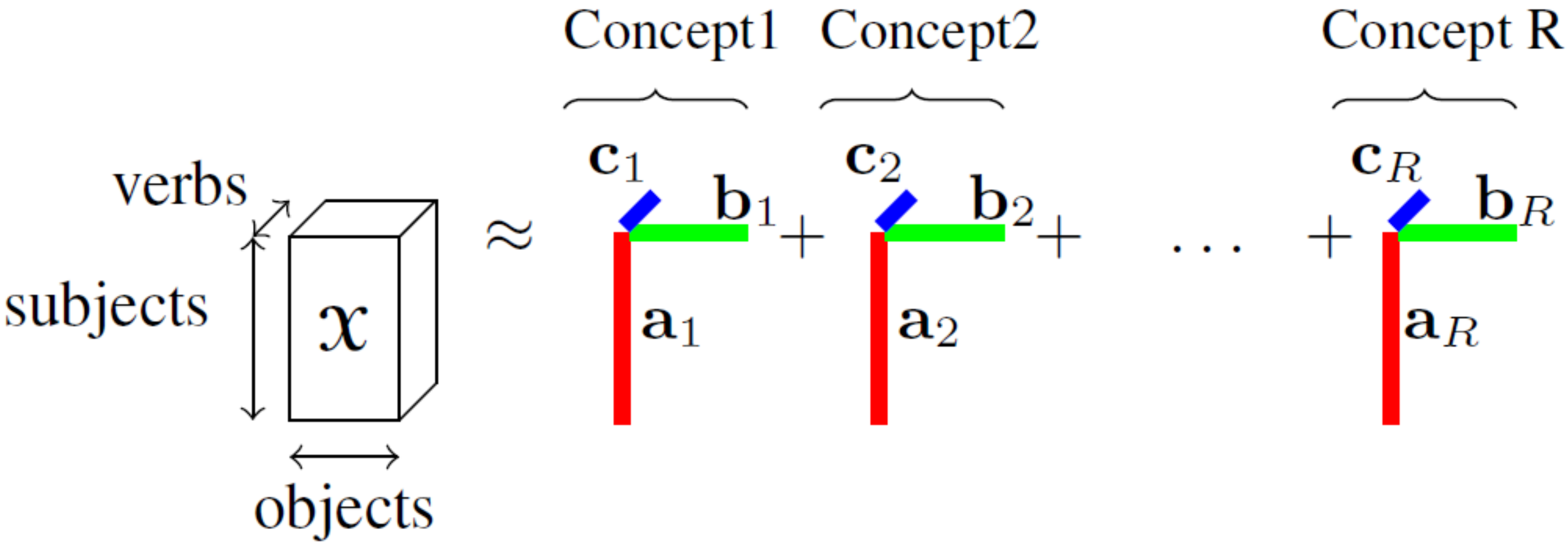
all I learned on tensors: from

Nikos Sidiropoulos
UMN

Tamara Kolda,
Sandia Labs
(tensor toolbox)

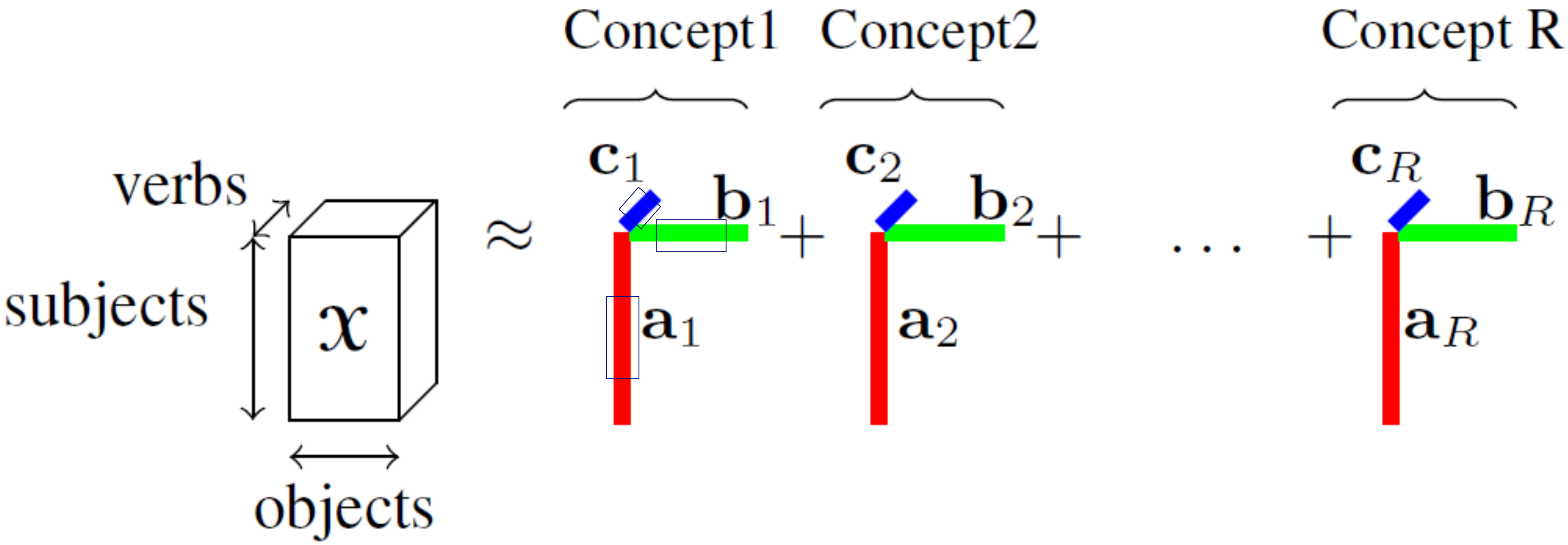
Problem Definition

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



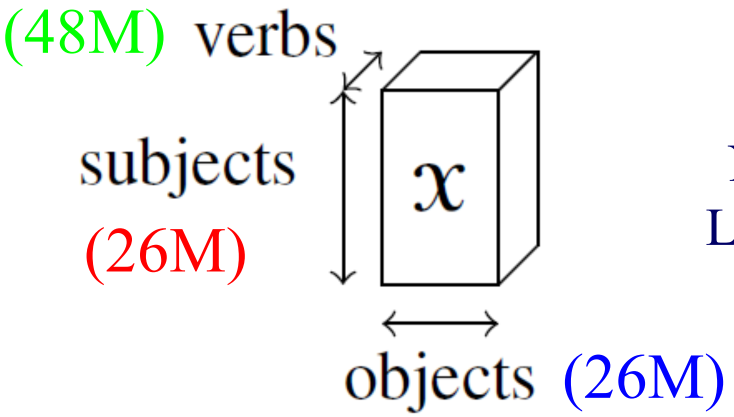
Problem Definition

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



Problem Definition

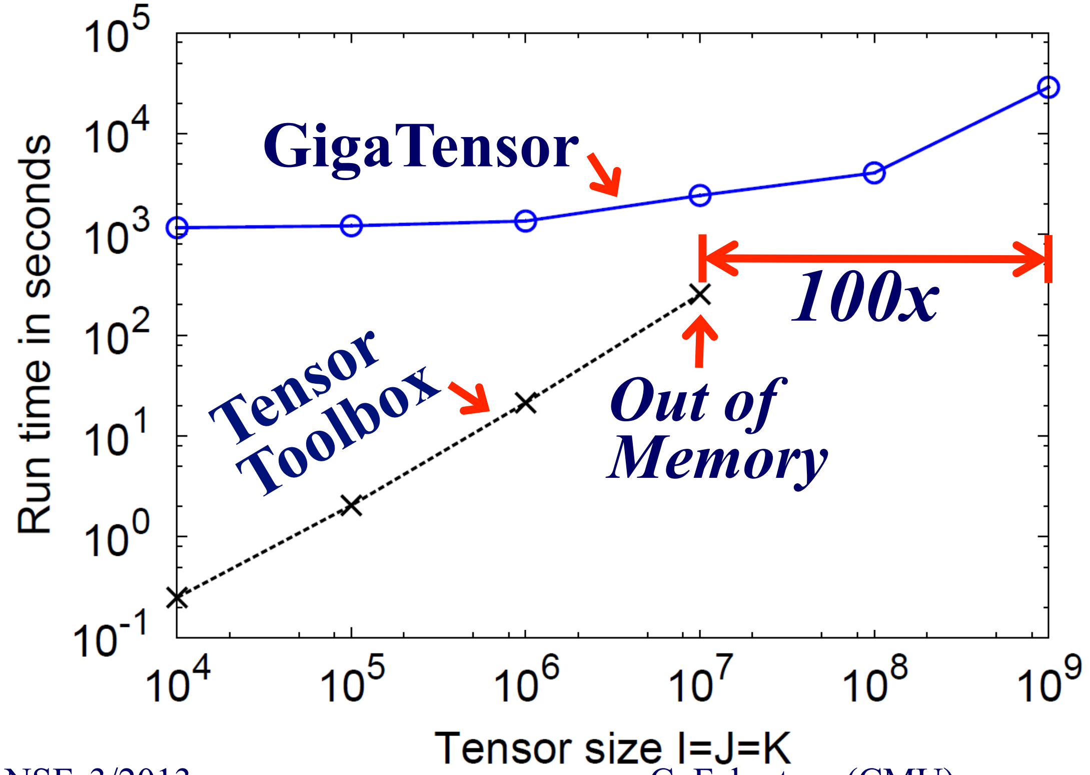
- Q1: Dominant concepts/topics?
- Q2: Find synonyms to a given noun phrase?
- (and how to scale up: |data| > RAM)

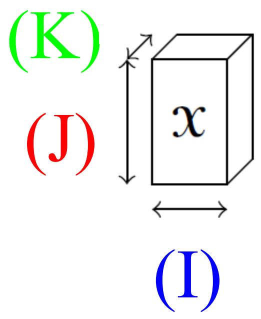


NELL (Never Ending Language Learner) data
Nonzeros =144M

Experiments

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

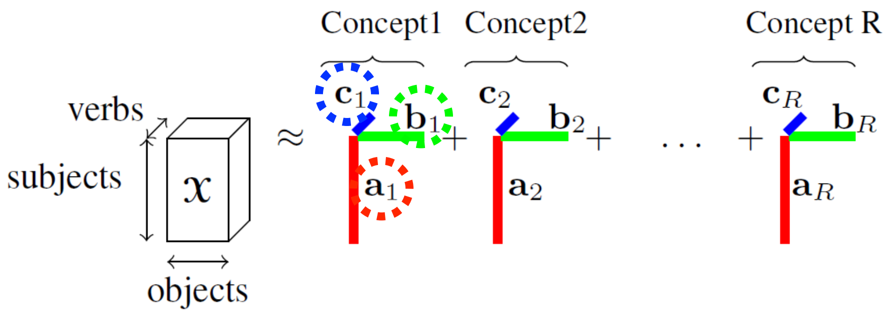




Number of nonzero = I / 50

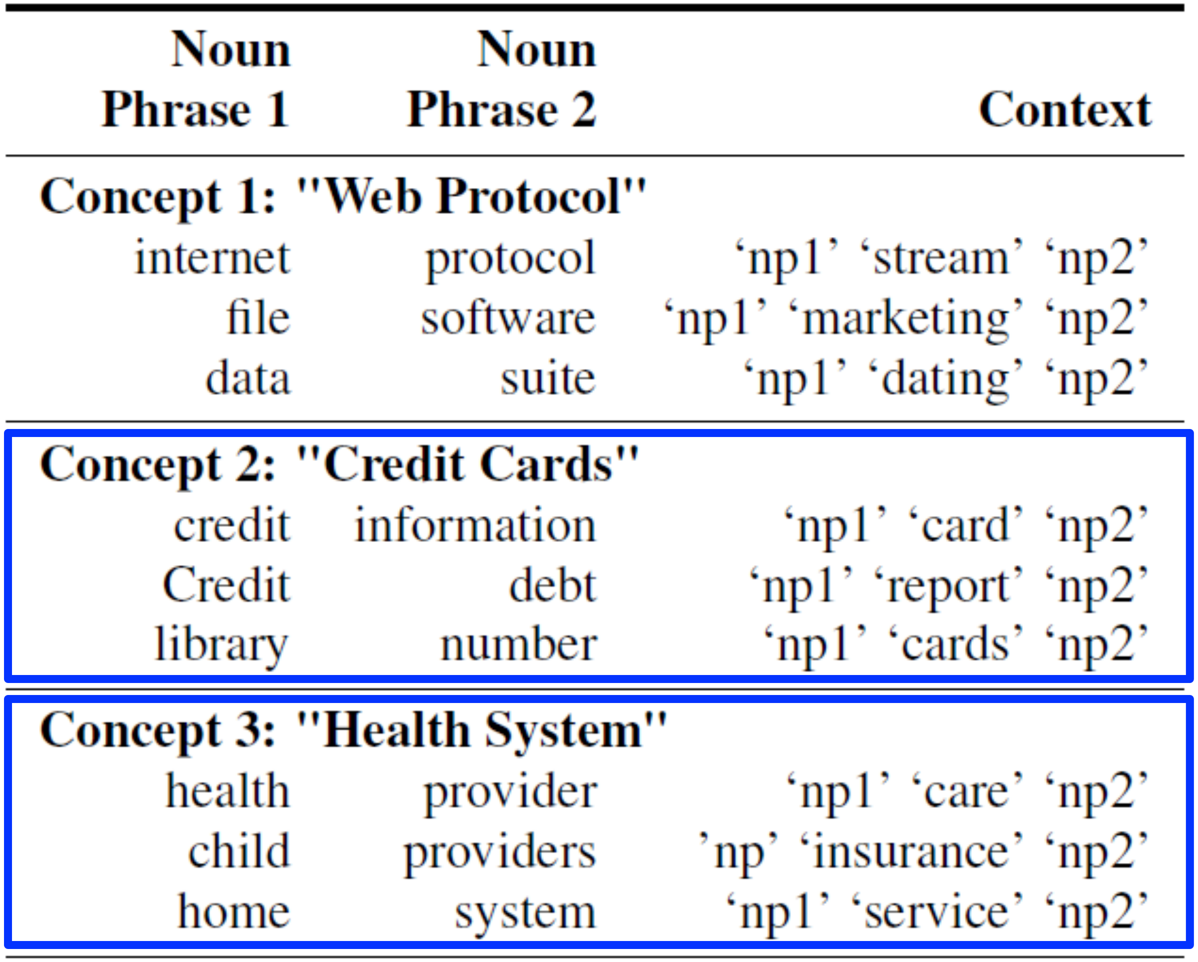
A1: Concept Discovery

- Concept Discovery in Knowledge Base



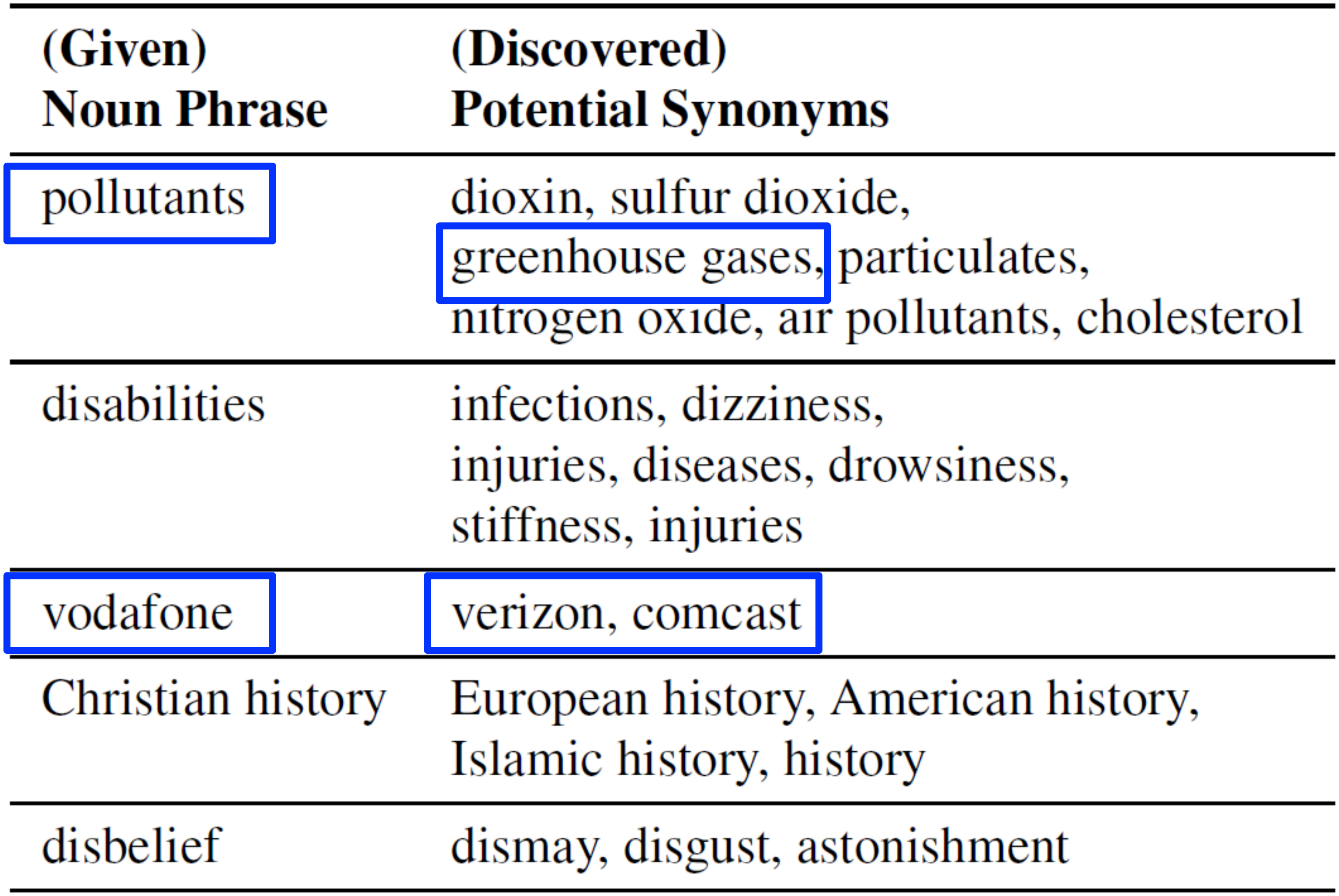
Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	‘np1’ ‘stream’ ‘np2’
file	software	‘np1’ ‘marketing’ ‘np2’
data	suite	‘np1’ ‘dating’ ‘np2’
Concept 2: "Credit Cards"		
credit	information	‘np1’ ‘card’ ‘np2’
Credit	debt	‘np1’ ‘report’ ‘np2’
library	number	‘np1’ ‘cards’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

A1: Concept Discovery



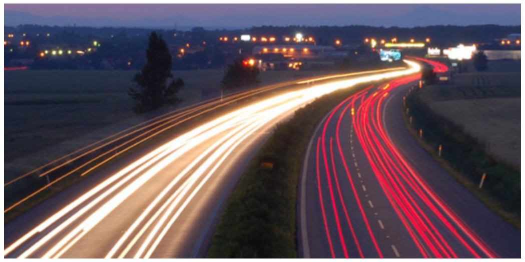
Noun Phrase 1	Noun Phrase 2	Context
Concept 1: "Web Protocol"		
internet	protocol	‘np1’ ‘stream’ ‘np2’
file	software	‘np1’ ‘marketing’ ‘np2’
data	suite	‘np1’ ‘dating’ ‘np2’
Concept 2: "Credit Cards"		
credit	information	‘np1’ ‘card’ ‘np2’
Credit	debt	‘np1’ ‘report’ ‘np2’
library	number	‘np1’ ‘cards’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

A2: Synonym Discovery



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

Roadmap

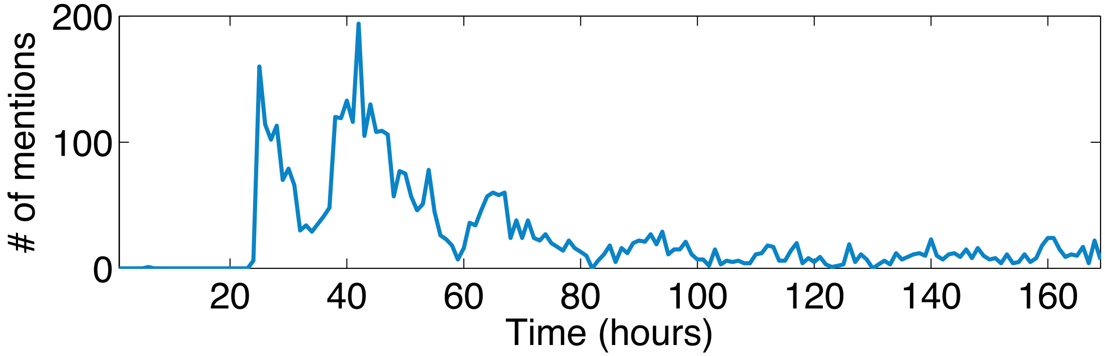


- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief propagation
  - Tensors
  - ➡ Spike analysis
  - Graph summarization
- Conclusions

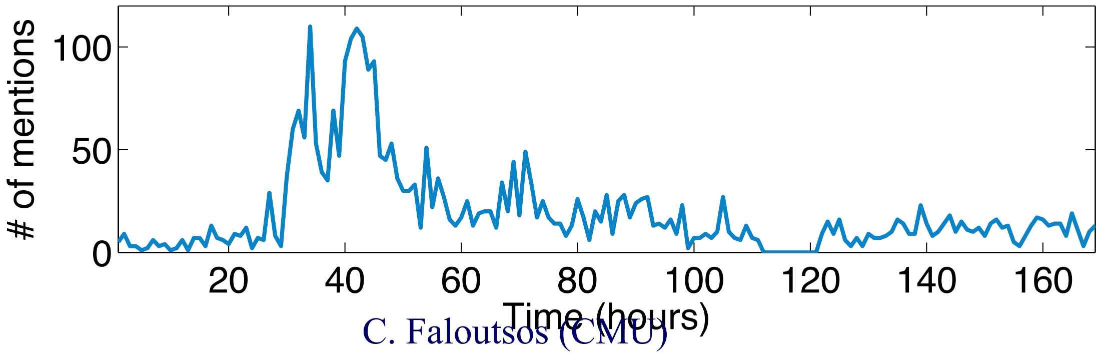
Rise and fall patterns in social media

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

“you can put lipstick on a pig”

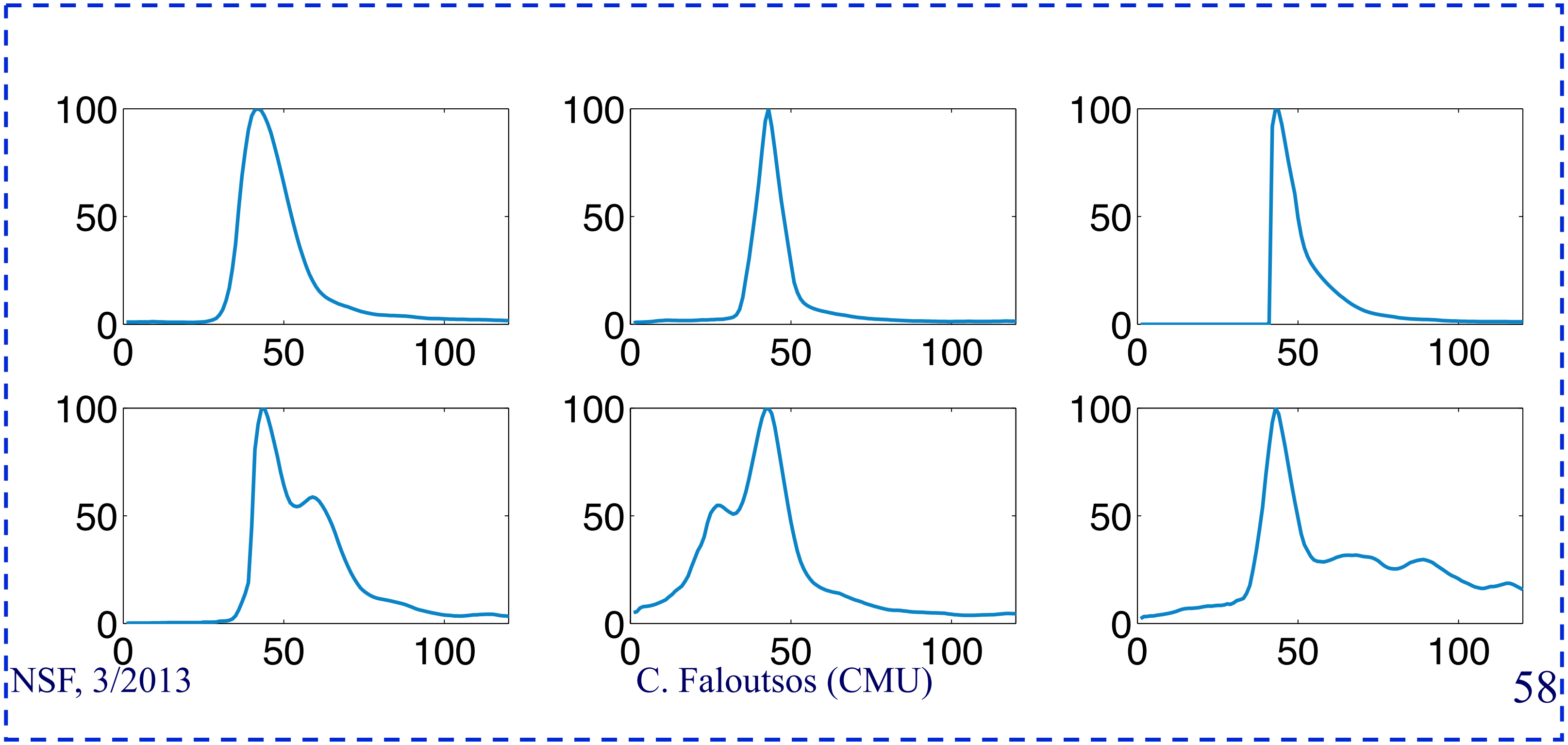


“yes we can”



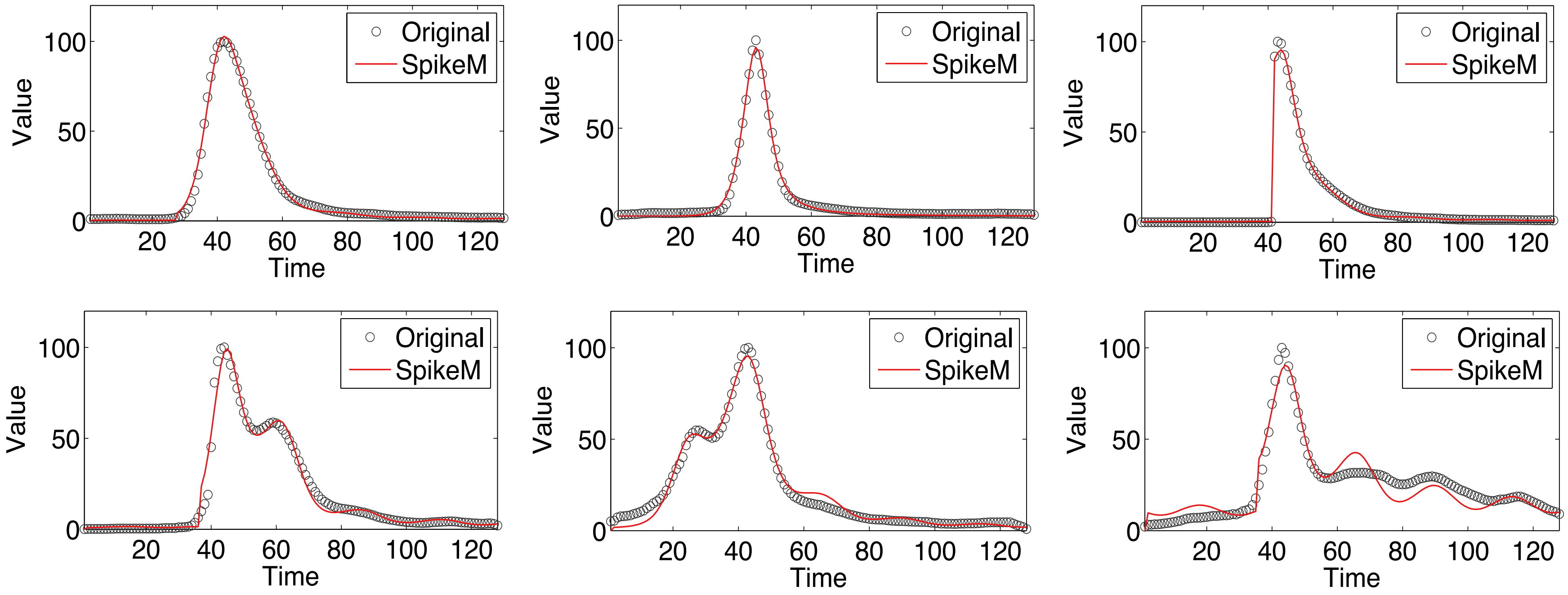
Rise and fall patterns in social media

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



Rise and fall patterns in social media

- Answer: YES!

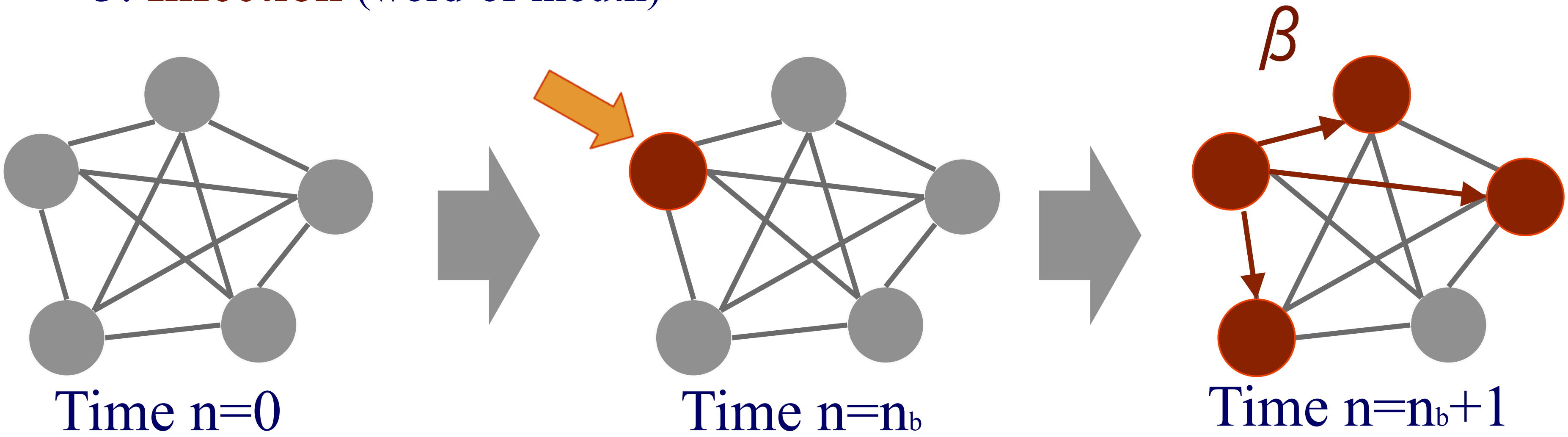


- We can represent **all patterns** by **single model**

In Matsubara+ SIGKDD 2012

Main idea - SpikeM

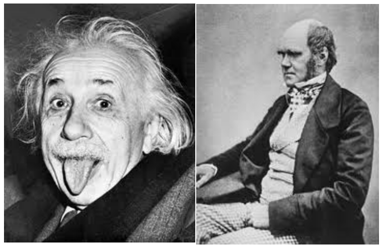
- 1. **Un**-informed bloggers (uninformed about rumor)
- 2. External shock at time n_b (e.g, breaking news)
- 3. Infection (word-of-mouth)



Infectiveness of a blog-post at age n:

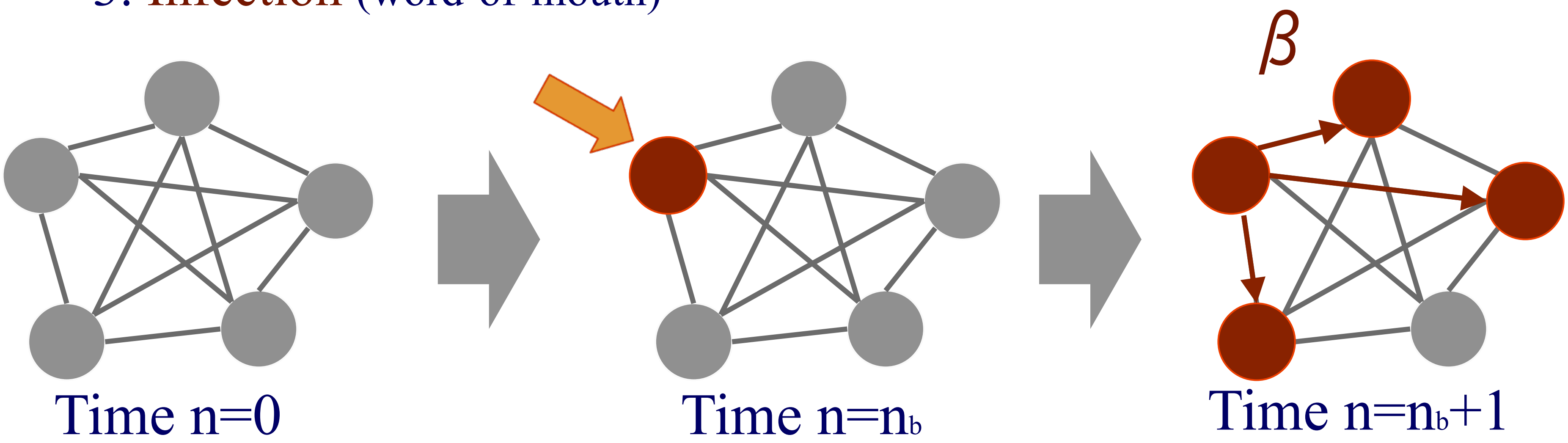
β – Strength of infection (quality of news)

$f(n)$ – Decay function



Main idea - SpikeM

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



Time n=0

Time $n=n_b$

Time $n=n_b+1$

Infectiveness of a blog-post at age n:

β – Strength of infection (quality of news)

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

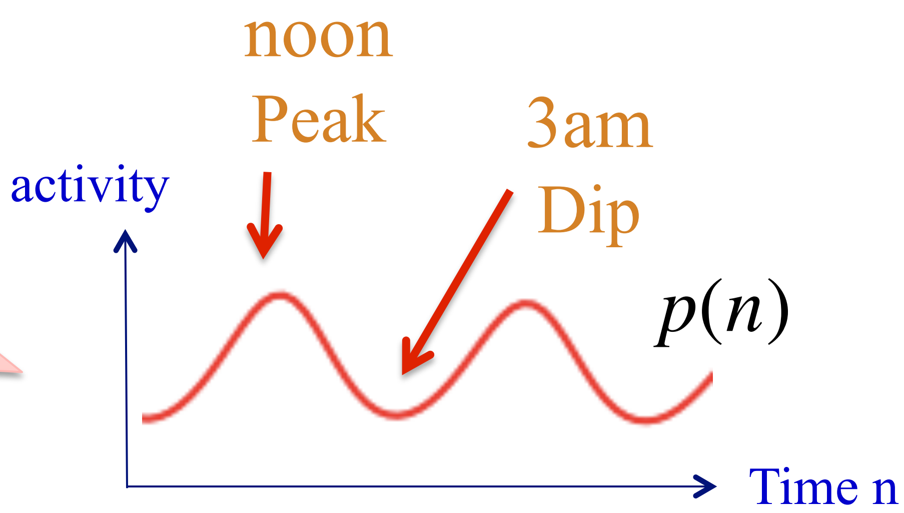
Details

SpikeM - with periodicity

- Full equation of SpikeM

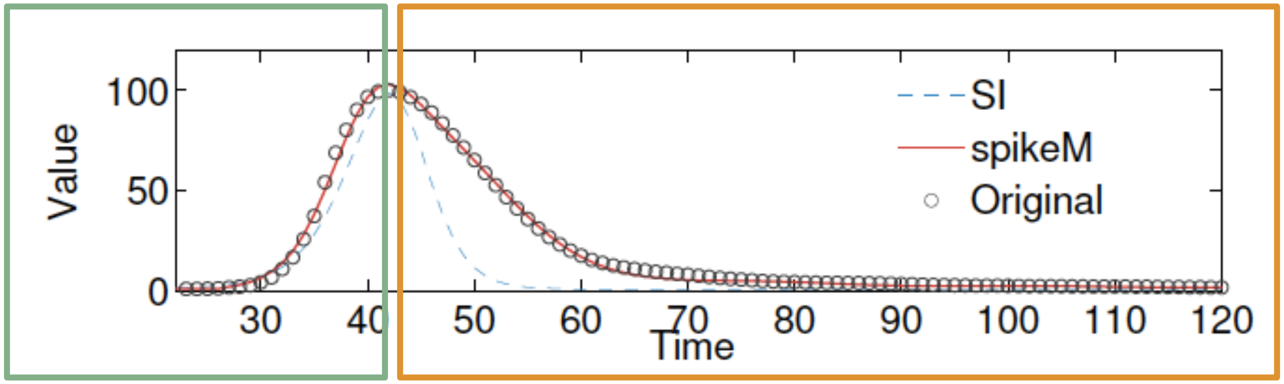
$$\Delta B(n+1) = \underset{\text{Periodicity}}{\underline{p(n+1)}} \cdot \left[U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot f(n+1-t) + \varepsilon \right]$$

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



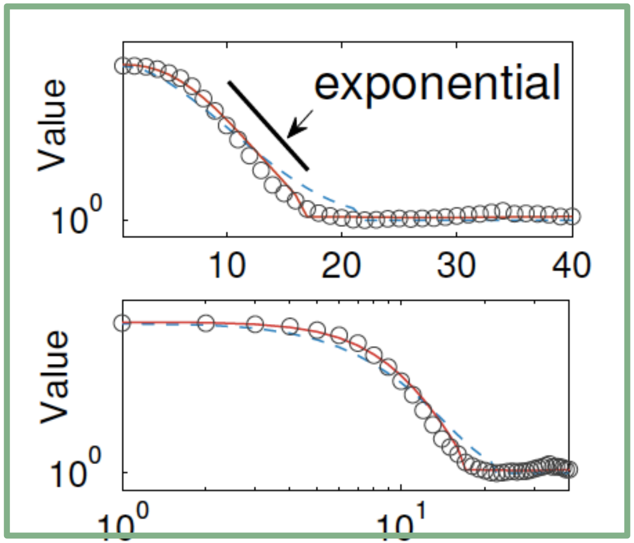
Details

- Analysis – exponential rise and power-raw fall



Lin-log

Log-log

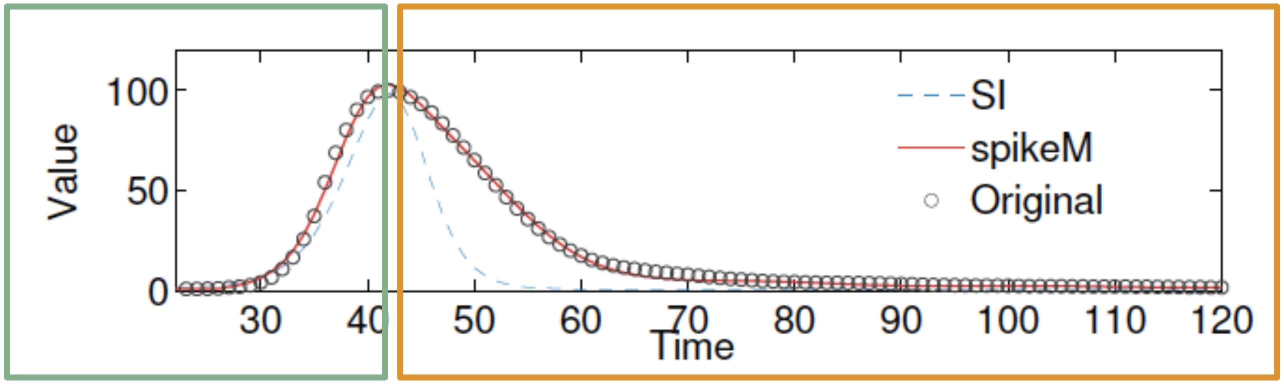


Rise-part

SI -> exponential
SpikeM -> exponential

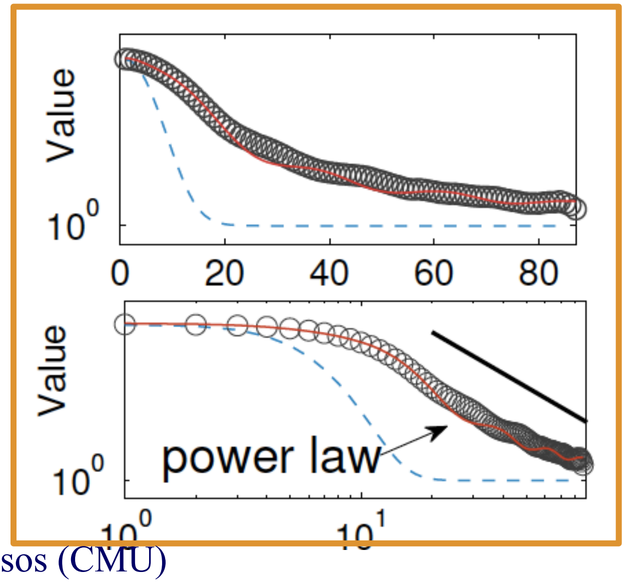
Details

- Analysis – exponential rise and power-raw fall



Fall-part

✘ SI -> exponential
SpikeM -> power law

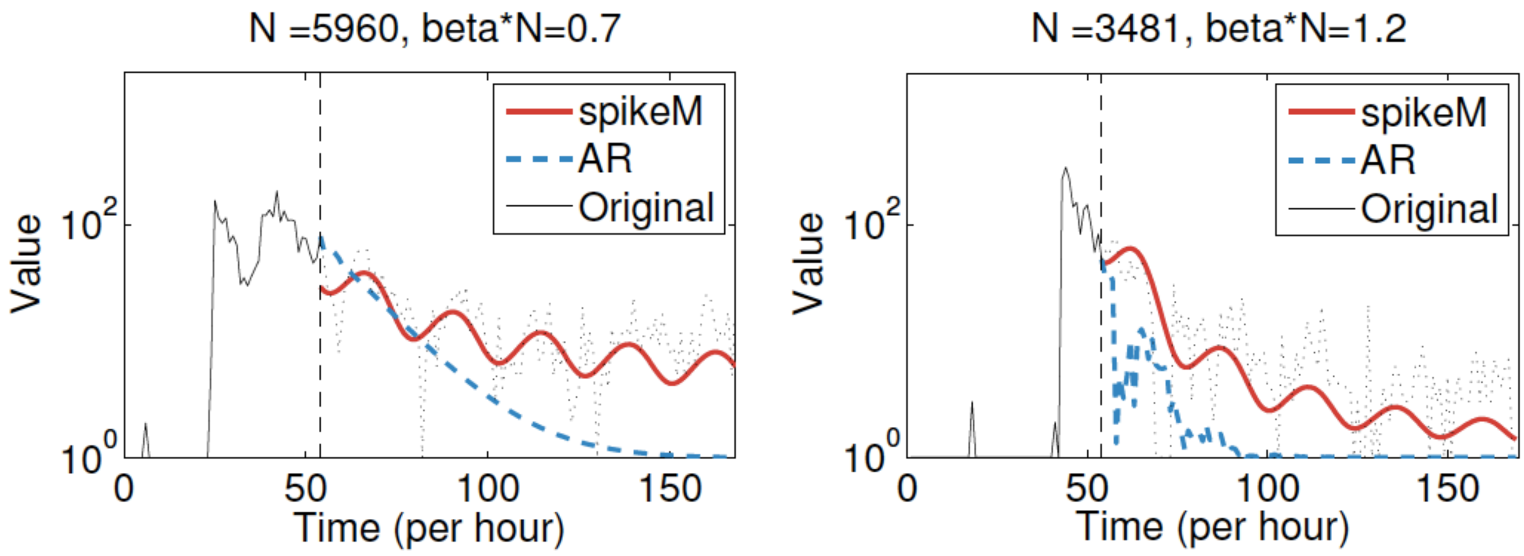


Lin-log

Log-log

Tail-part forecasts

- **SpikeM** can capture tail part

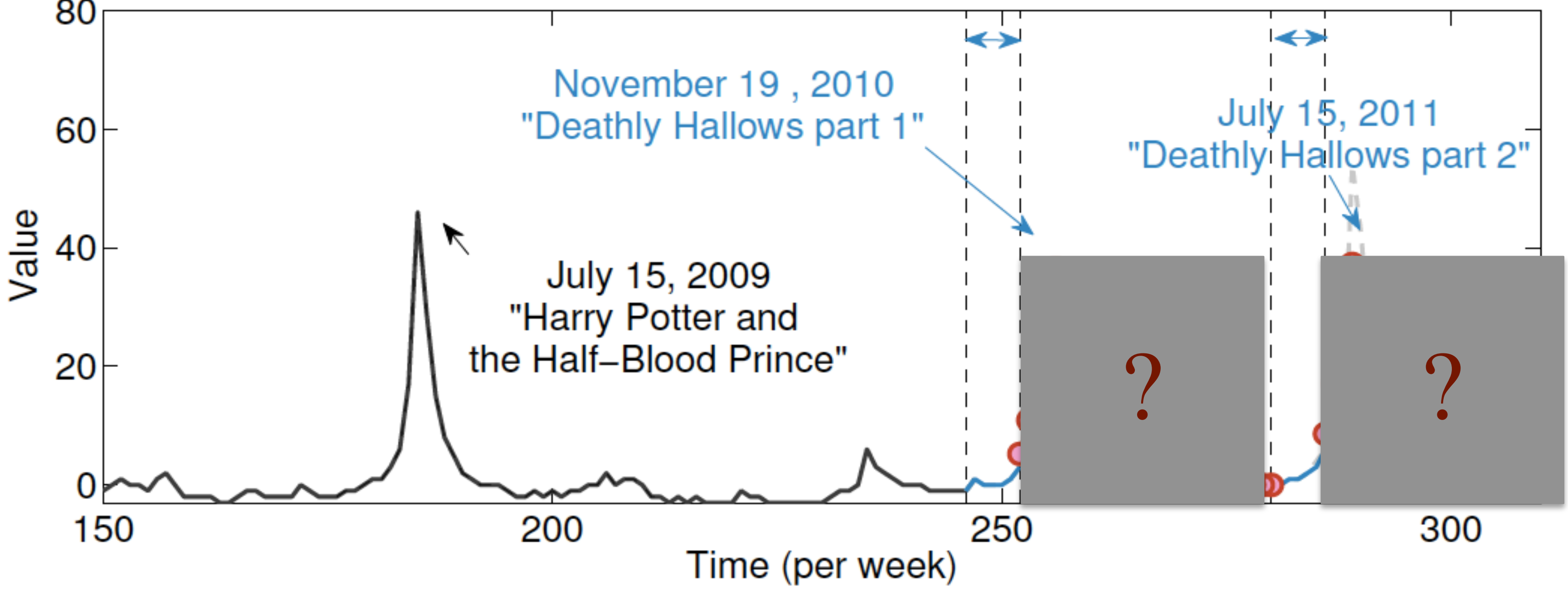


"What-if" forecasting

(1) First spike

(2) Release date

(3) Two weeks before release



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

(2) release date of two sequel movies

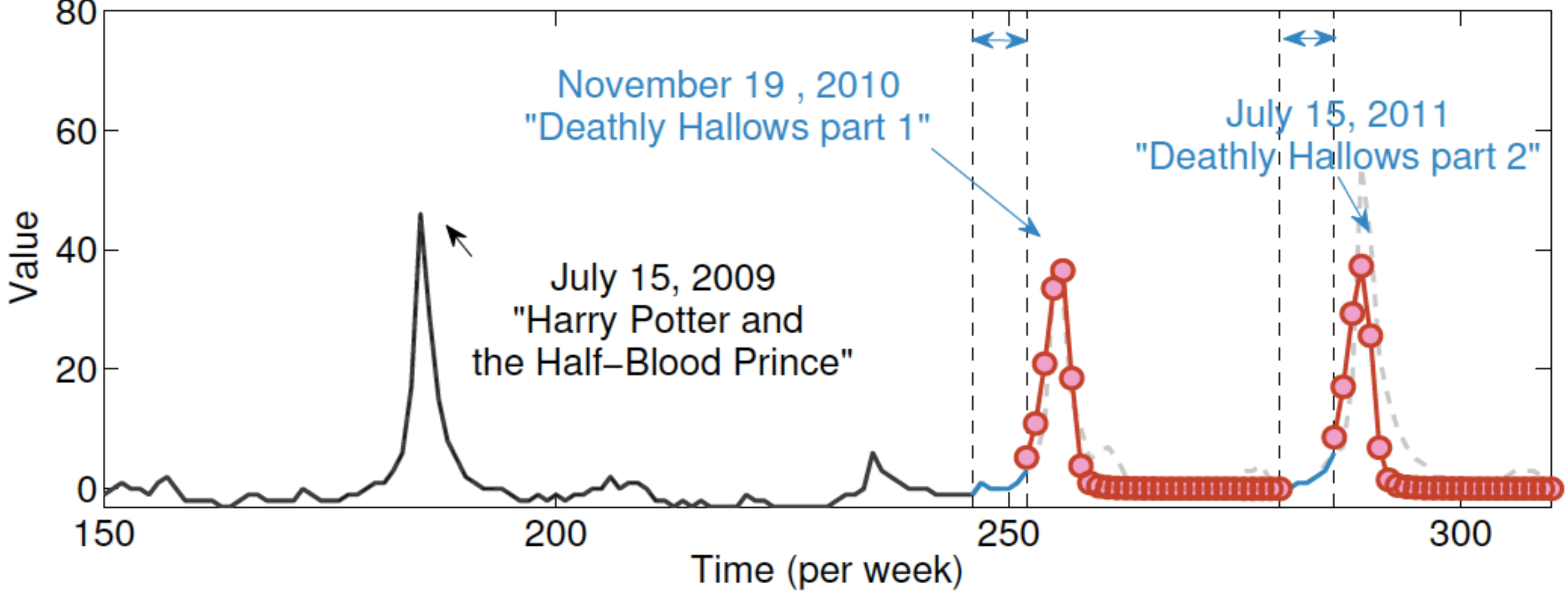
(3) access volume before the release date

“What-if” forecasting

(1) First spike

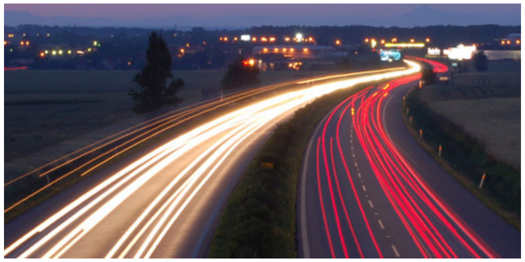
(2) Release date

(3) Two weeks before release



SpikeM can forecast upcoming spikes

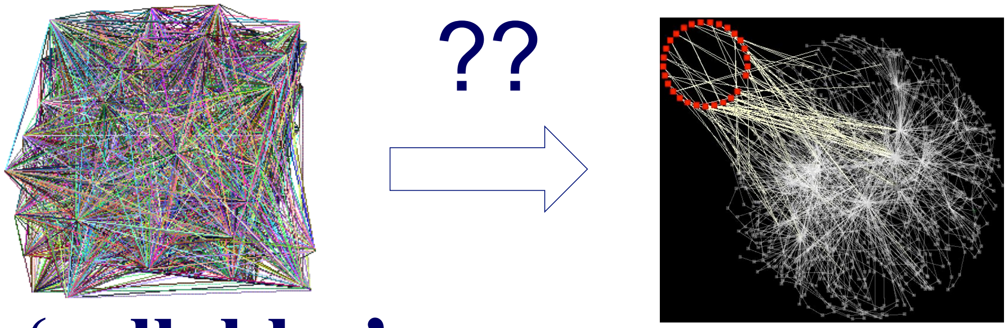
Roadmap



- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
  - Belief Propagation
  - Tensors
  - Spike analysis
  - ➡ Graph understanding (through MDL)
- Conclusions

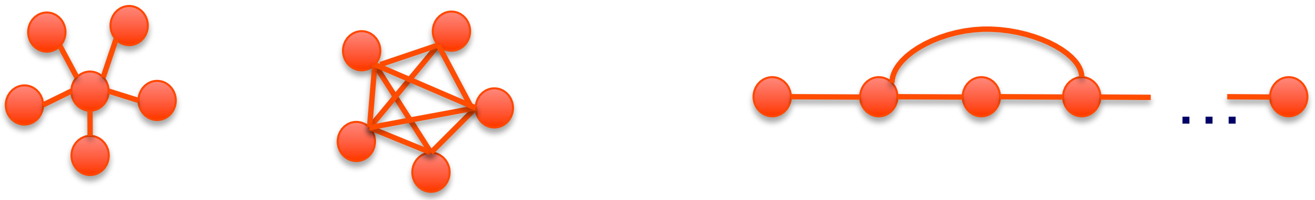
Summarizing Graphs

Goal:



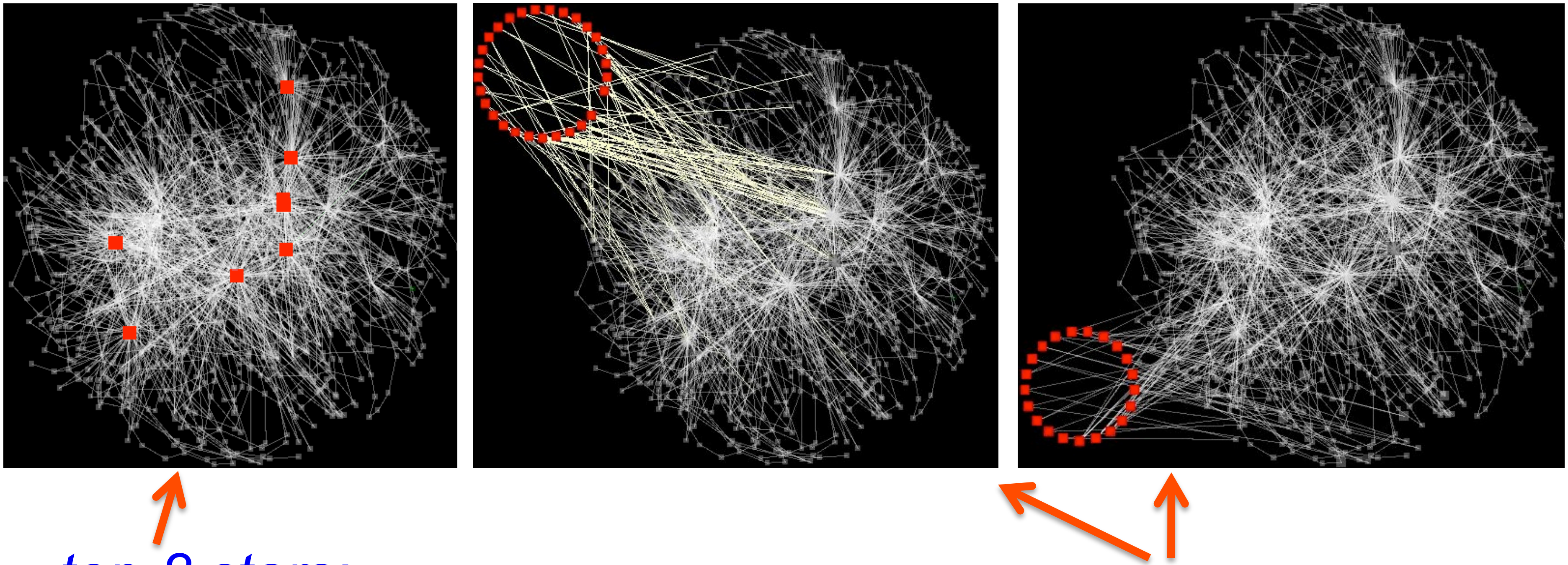
Main Idea: MDL + ‘syllables’ :

star, clique, chain, bi-partite core



Koutra, Kang, Vreeken, et al, (subm.)

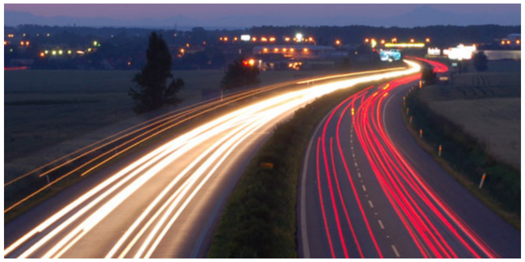
Summarizing Wiki-controversy



top-8 stars: admins, bots

top-1 and top-2 bipartite cores: edit wars.
***Left**:* warring factions ('Kiev' vs 'Kyev')
***Right**: between vandals*

Roadmap



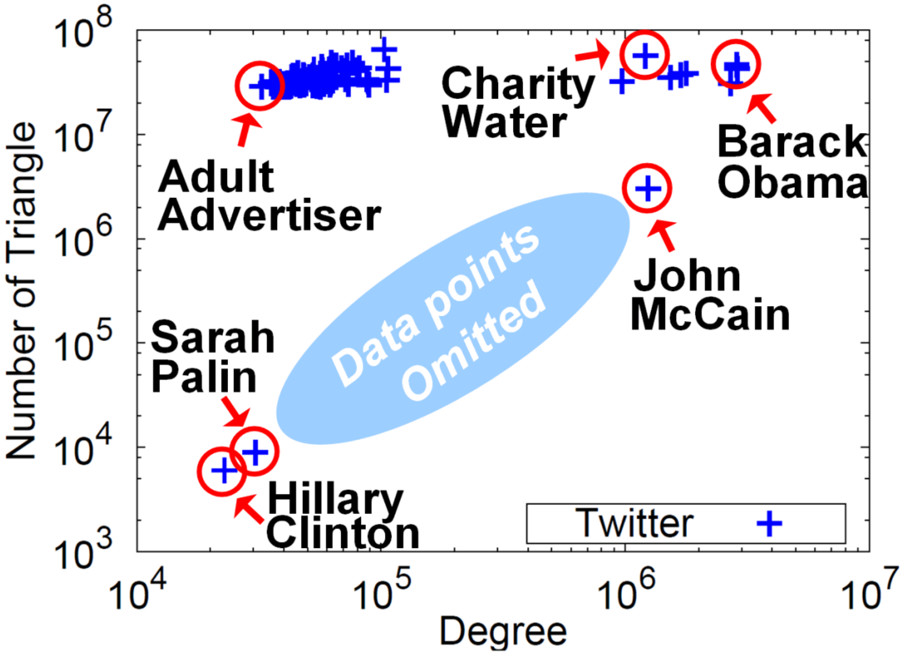
- Introduction – Motivation
- Problem#1: Patterns in graphs
- Problem#2: Tools
- ➡ Conclusions

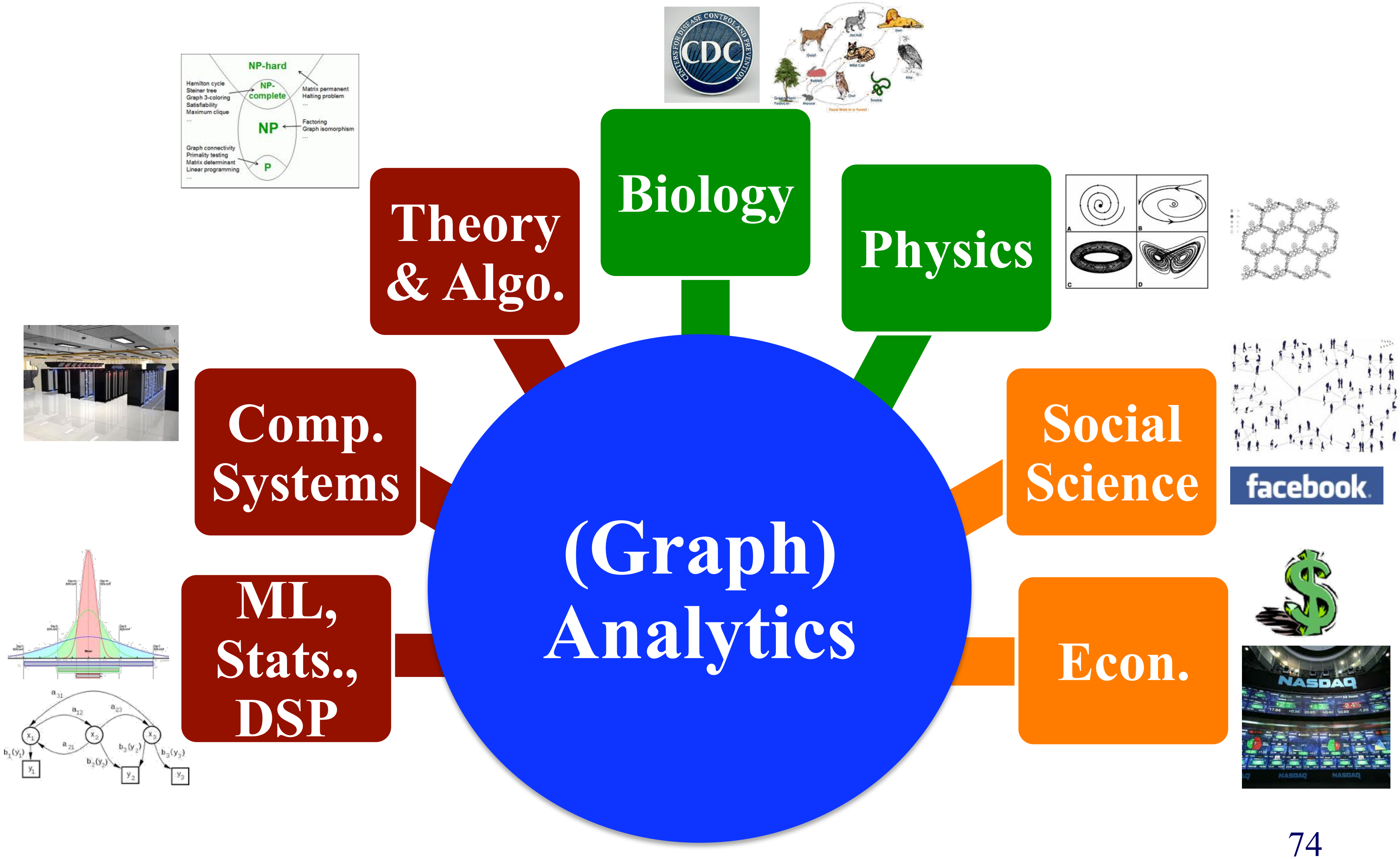
OVERALL CONCLUSIONS – low level:

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

OVERALL CONCLUSIONS – high level

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





Theory & Algo.

Biology

Physics

Comp. Systems

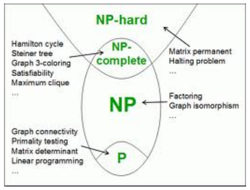
Social Science

ML, Stats., DSP

Econ.

(Graph) Analytics

Cross-disciplinarity: A must

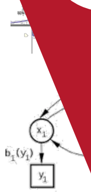


Theory & Algo

Biology

C

al Science



)

alytics

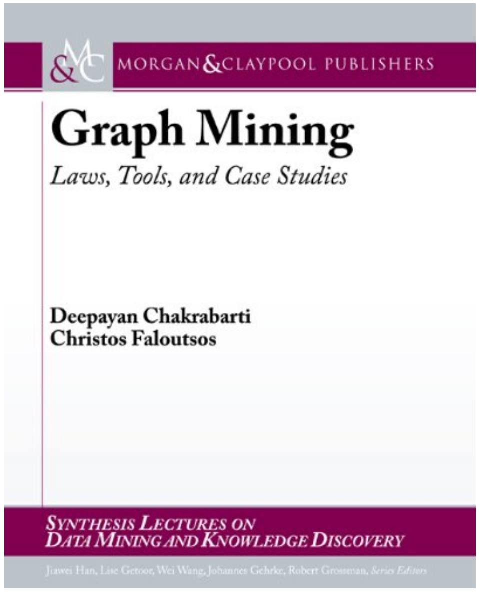
Econ.

References

- Leman Akoglu, Christos Faloutsos: *RTG: A Recursive Realistic Graph Generator Using Random Typing*. ECML/PKDD (1) 2009: 13-28
- Deepayan Chakrabarti, Christos Faloutsos: *Graph mining: Laws, generators, and algorithms*. ACM Comput. Surv. 38(1): (2006)

References

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



&C MORGAN & CLAYPOOL PUBLISHERS

Graph Mining

Laws, Tools, and Case Studies

Deepayan Chakrabarti
Christos Faloutsos

SYNTHESIS LECTURES ON
DATA MINING AND KNOWLEDGE DISCOVERY

Jiawei Han, Lise Getoor, Wei Wang, Johannes Gehrke, Robert Grossman, *Series Editors*

References

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

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

References

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

References

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

References

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

References

- Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, Christos Faloutsos, "*Rise and Fall Patterns of Information Diffusion: Model and Implications*", KDD'12, pp. 6-14, Beijing, China, August 2012

References

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

References

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

References

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

References

- Hanghang Tong, Christos Faloutsos, Brian Gallagher, Tina Eliassi-Rad: *Fast best-effort pattern matching in large attributed graphs*. KDD 2007: 737-746
- *(Best paper* award, CIKM'12) Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos and Christos Faloutsos Gelling, and Melting, Large Graphs by Edge Manipulation, Maui, Hawaii, USA, Oct. 2012.

References

- Hanghang Tong, Spiros Papadimitriou, Christos Faloutsos, Philip S. Yu, Tina Eliassi-Rad: Gateway finder in large graphs: problem definitions and fast solutions. Inf. Retr. 15(3-4): 391-411 (2012)

Project info & ‘thanks’

www.cs.cmu.edu/~pegasus

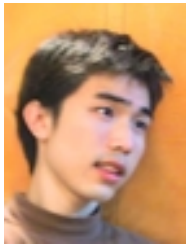
PROJECT PEGASUS

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

Cast

Akoglu, Leman

Beutel, Alex



Chau, Polo

Kang, U

Koutra, Danai

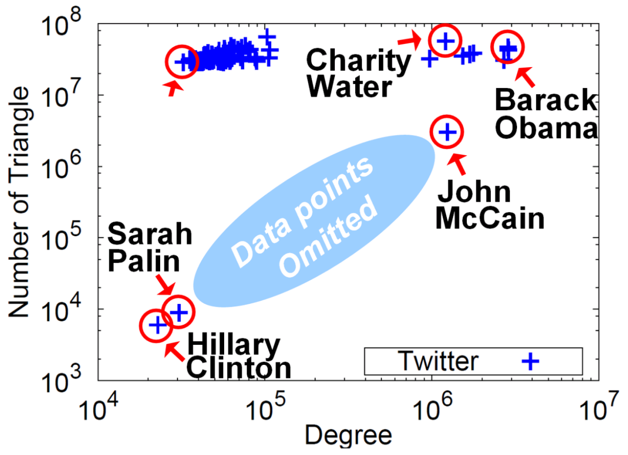
McGlohon, Mary

Prakash, Aditya

Papalexakis, Vagelis

Tong, Hanghang

Take-home message



Tera/Peta-byte data ⇨ Analytics ⇨ Insights, outliers

Big data reveal **insights** that would be invisible otherwise (even to **experts**)