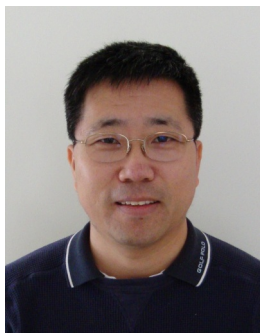


Mining Large Graphs

Christos Faloutsos

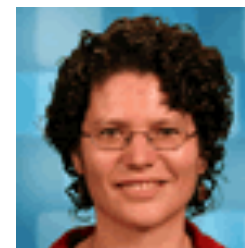
CMU

Thank you!



Prof. Bing Liu

Prof. Tanya Berger-Wolf



Kitty Ngyuen

Hi!



Prof. Isabel Cruz



Prof. Clement Yu



Prof. Jakob Eriksson



Prof. Philip Yu



Prof. Brian Ziebart



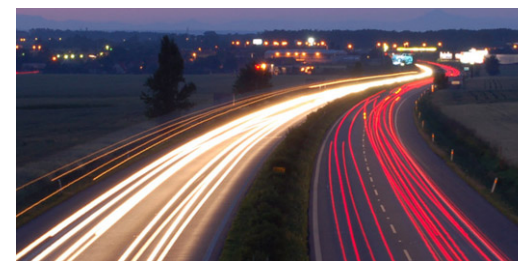
Prof. Liz Marai



Prof. Ouri Wolfson

Roadmap

- ➔ • Introduction – Motivation
 - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Conclusions

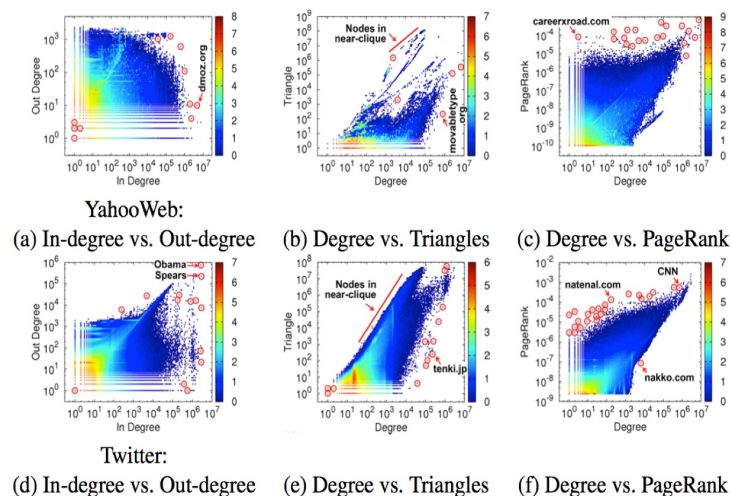


Graphs - why should we care?



~1B nodes (web sites)
~6B edges (http links)
'YahooWeb graph'

Graphs - why should we care?



~1B nodes (web sites)
 ~6B edges (http links)
 'YahooWeb graph'

U Kang, Jay-Yoon Lee, Danai Koutra, and Christos Faloutsos. *Net-Ray: Visualizing and Mining Billion-Scale Graphs* PAKDD 2014, Tainan, Taiwan.

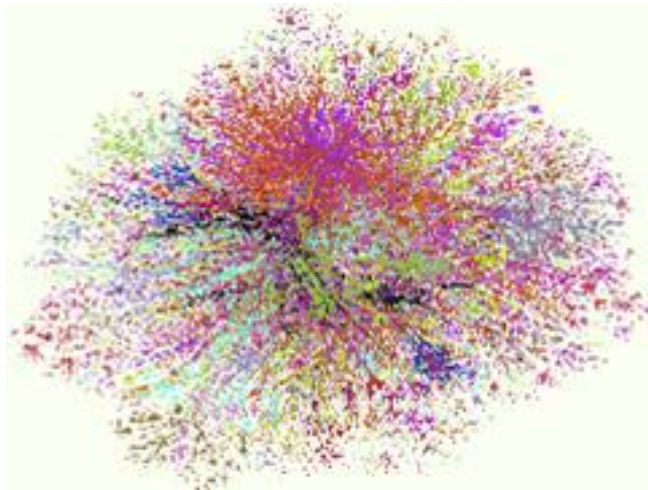
Graphs - why should we care?



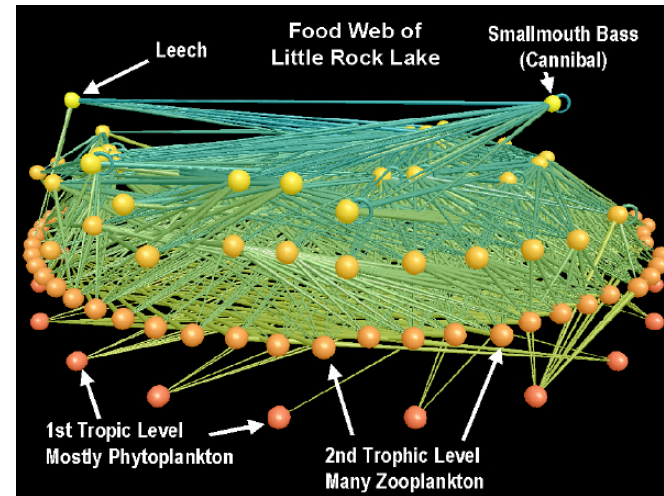
>\$10B; ~1B users



Graphs - why should we care?



Internet Map
[lumeta.com]



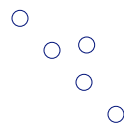
Food Web
[Martinez '91]

Graphs - why should we care?

- web-log ('blog') news propagation 
- computer network security: email/IP traffic and anomaly detection
- Recommendation systems 
-
- Many-to-many db relationship -> graph

Motivating problems

- P1: patterns? Fraud detection?



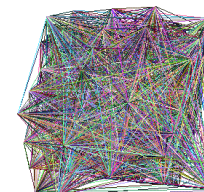
- P2: patterns in time-evolving graphs / tensors

destination



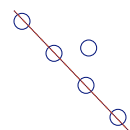
source

time



Motivating problems

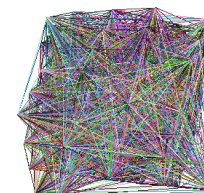
- P1: patterns? Fraud detection?



Patterns



anomalies



- P2: patterns in time-evolving graphs / tensors

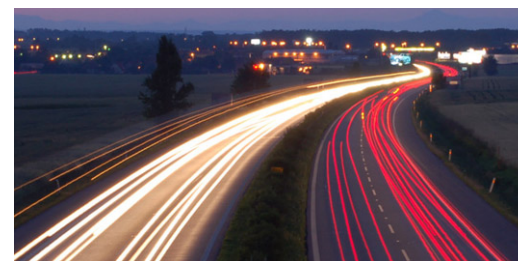
destination



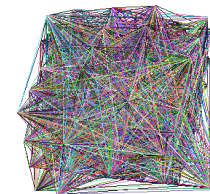
source

time

Roadmap



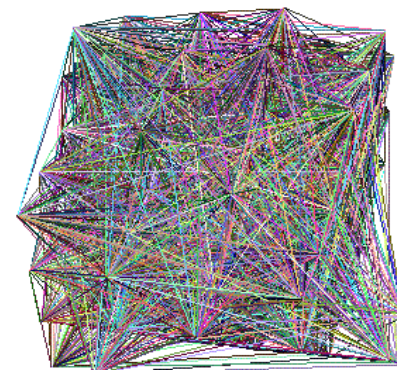
- Introduction – Motivation
 - Why study (big) graphs?
- ➔ • Part#1: Patterns & fraud detection
- Part#2: time-evolving graphs; tensors
- Conclusions



Part 1: Patterns, & fraud detection

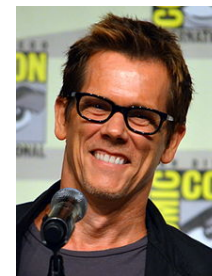
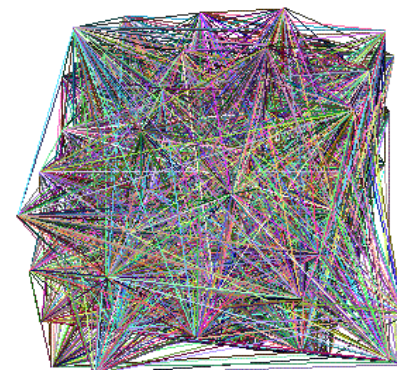
Laws and patterns

- Q1: Are real graphs random?



Laws and patterns

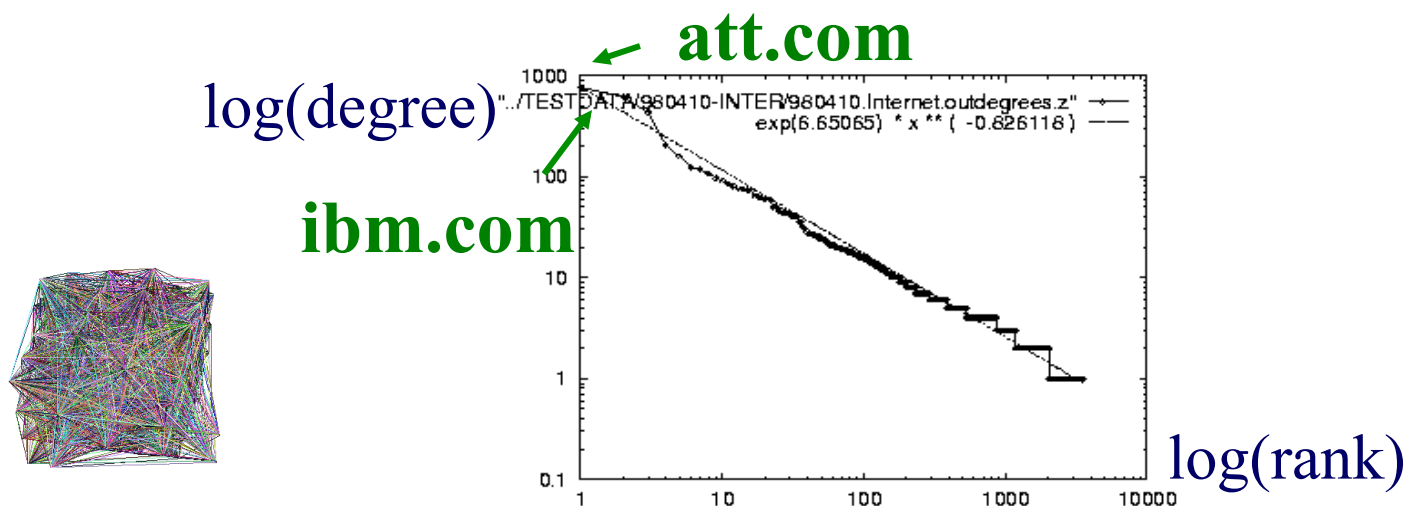
- Q1: Are real graphs random?
- A1: NO!!
 - Diameter ('6 degrees'; 'Kevin Bacon')
 - in- and out- degree distributions
 - other (surprising) patterns
- So, let's look at the data



Solution# S.1

- Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

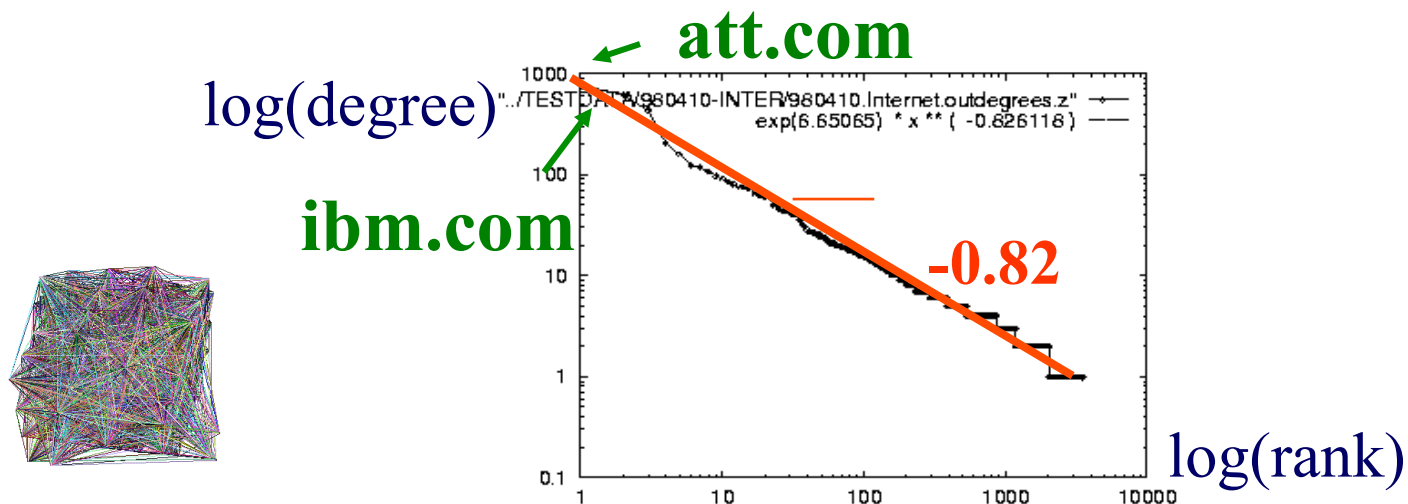
internet domains



Solution# S.1

- Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

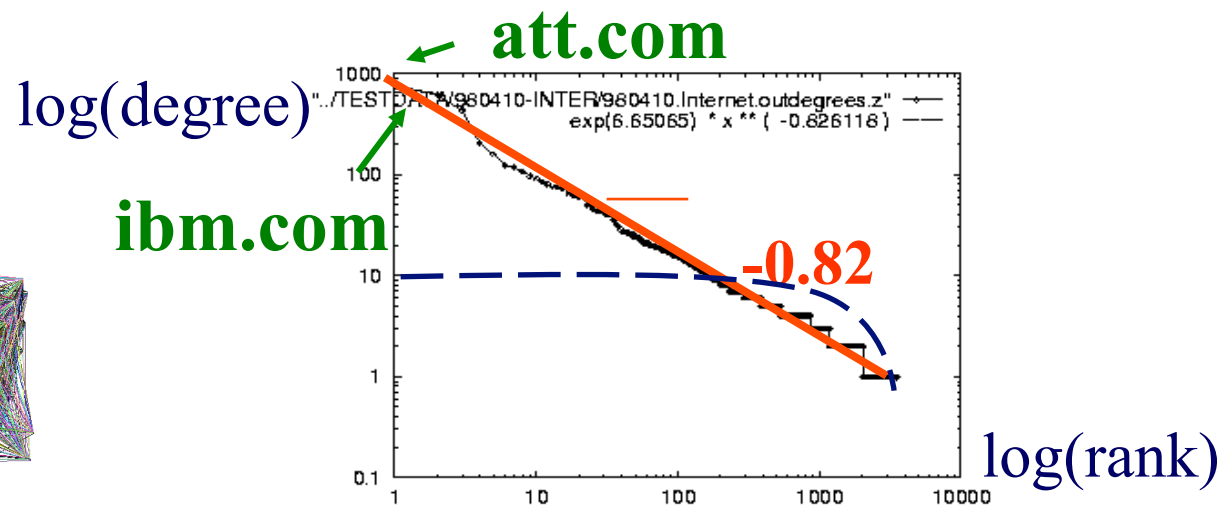
internet domains



Solution# S.1

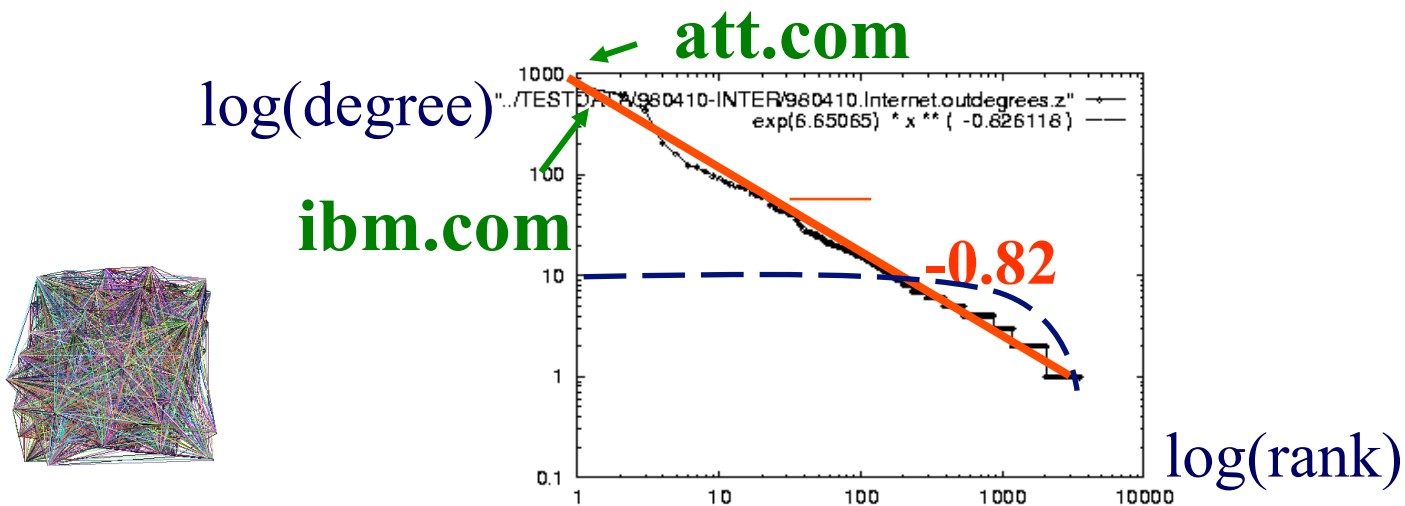
- Q: So what?

internet domains



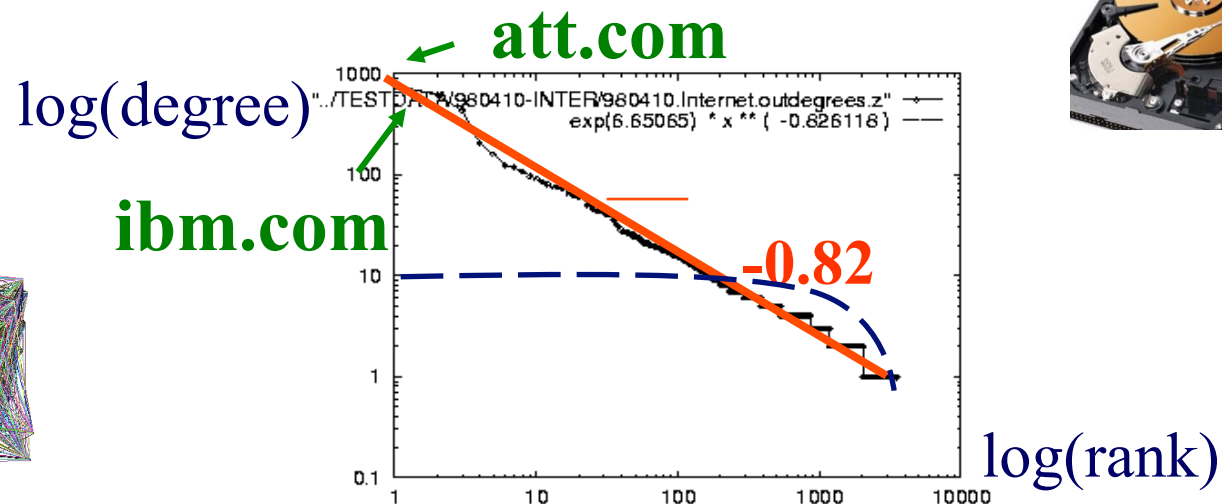
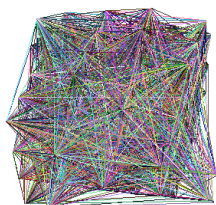
Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs:
internet domains



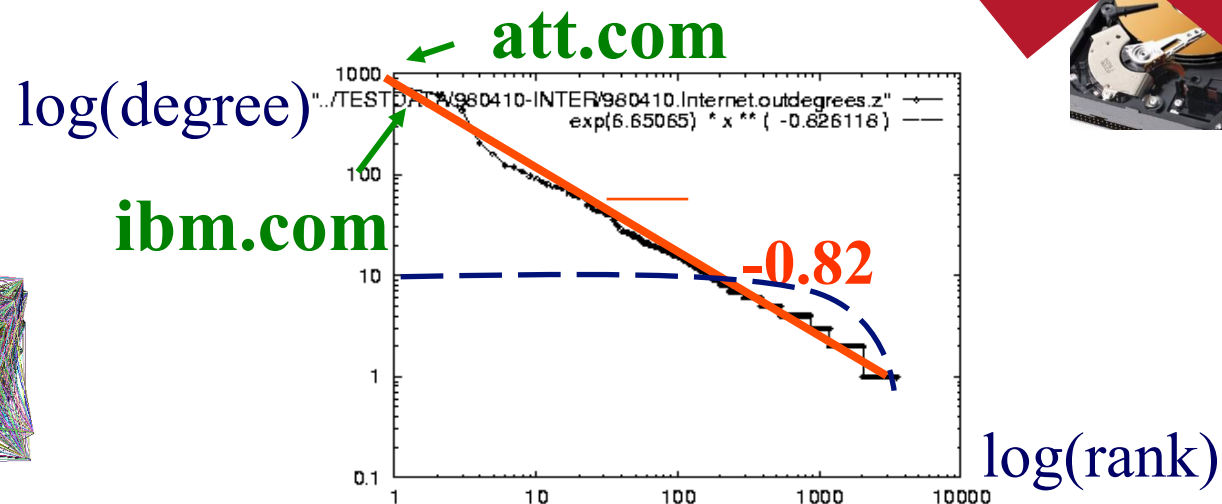
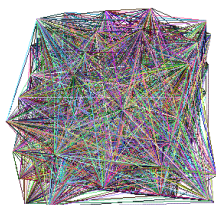
Solution# S.1

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



Solution# S.1

- Q: So what?
- A1: # of two-step-away pairs: $100^2 \times 100$ = friends of friends (F.O.F.)
internet domains



Gaussian trap

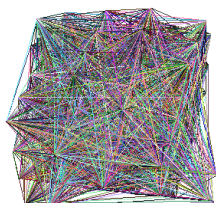
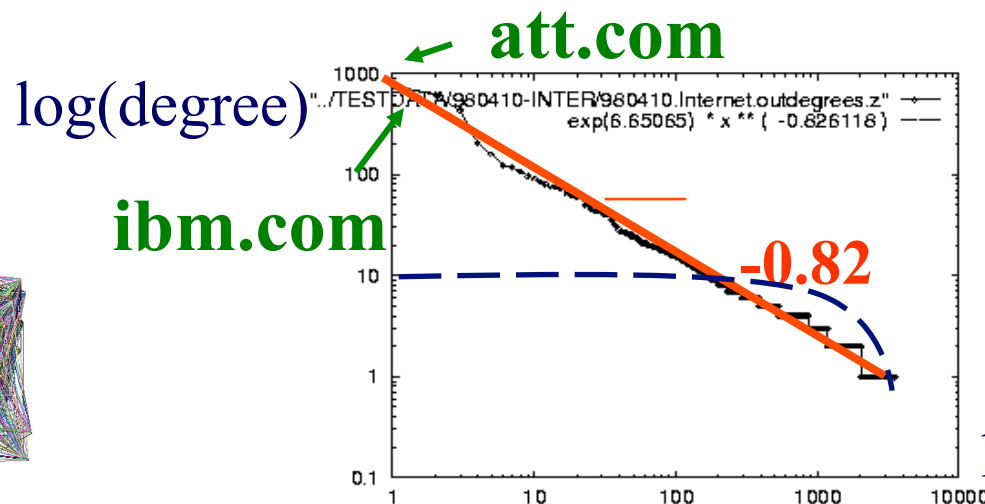
Solution# S.1



- Q: So what?
- A1: # of two-step-away pairs: $O(d_{\max}^2) \sim 10M^2$
internet domains



$\sim 0.8\text{PB} \rightarrow$
a data center(!)



Gaussian trap

Solution# S.1

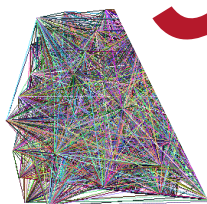


- Q: So what?
- A1: # of two-step-away
inter

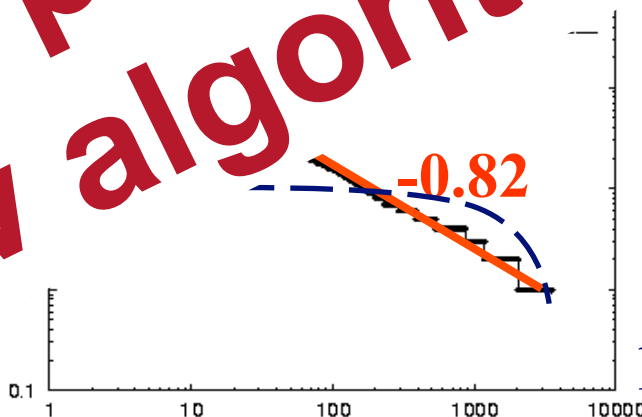
$$?) \sim 10M^2$$



$\sim 0.8PB \rightarrow$
a data center(!)

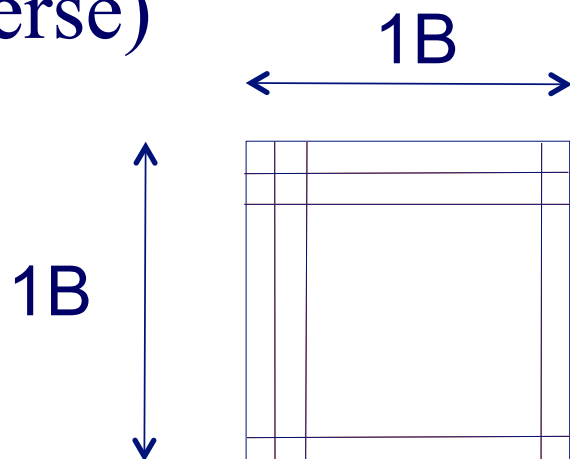


**Such patterns $\rightarrow$
New algorithms**



Observation – big-data:

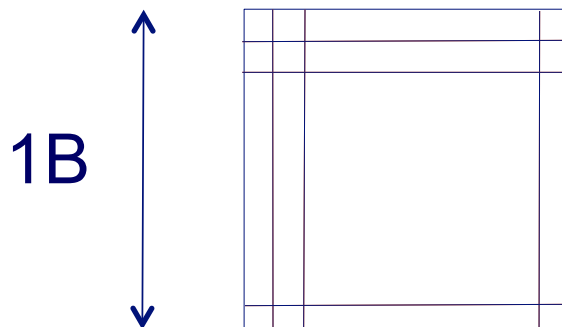
- $O(N^2)$ algorithms are $\sim$ intractable - $N=1\text{B}$
- N^2 seconds = 31B years ($>2\times$ age of universe)



Observation – big-data:

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

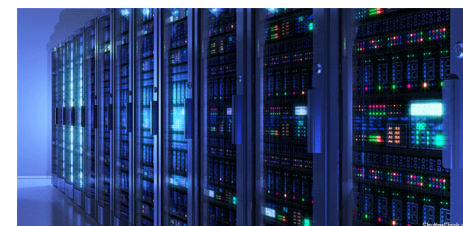
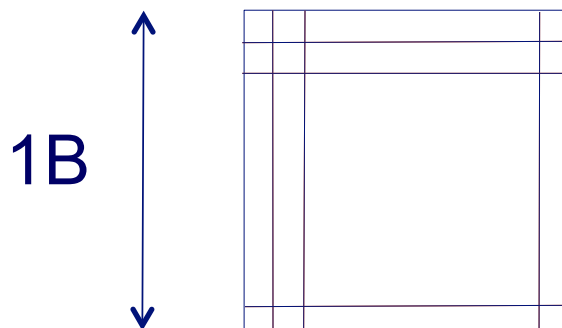
- N^2 seconds = ~~31B~~ ^{31M} years
- 1,000 machines



Observation – big-data:

- $O(N^2)$ algorithms are $\sim$ intractable - $N=1B$

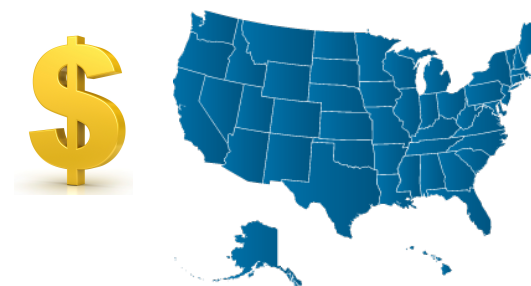
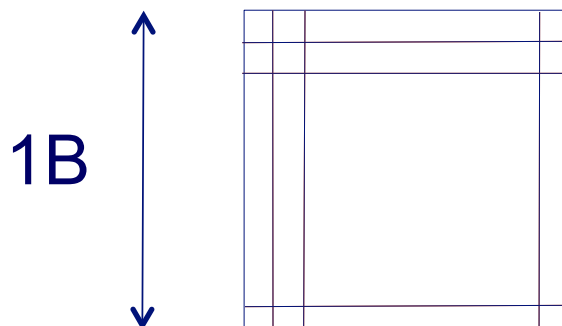
- N^2 seconds = ~~31B~~ ^{31K} years
- 1M machines



Observation – big-data:

- $O(N^2)$ algorithms are ~intractable - $N=1\text{B}$

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

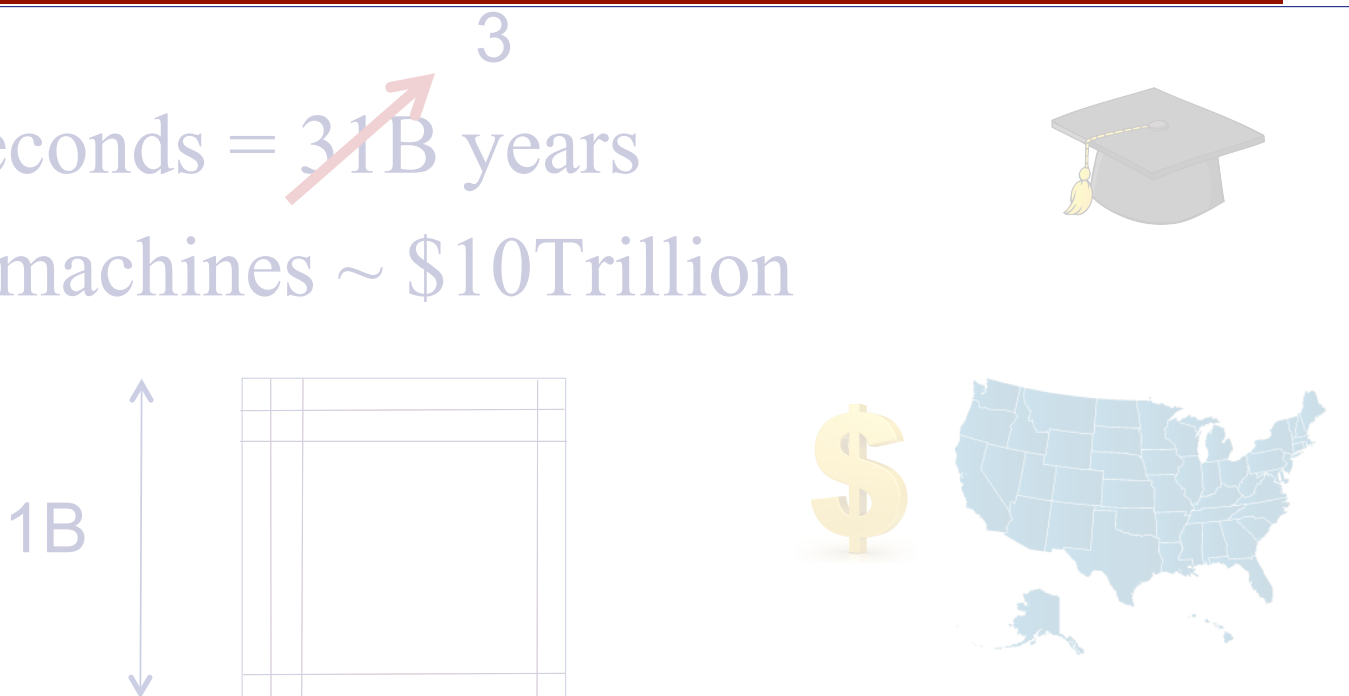


Observation – big-data:

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

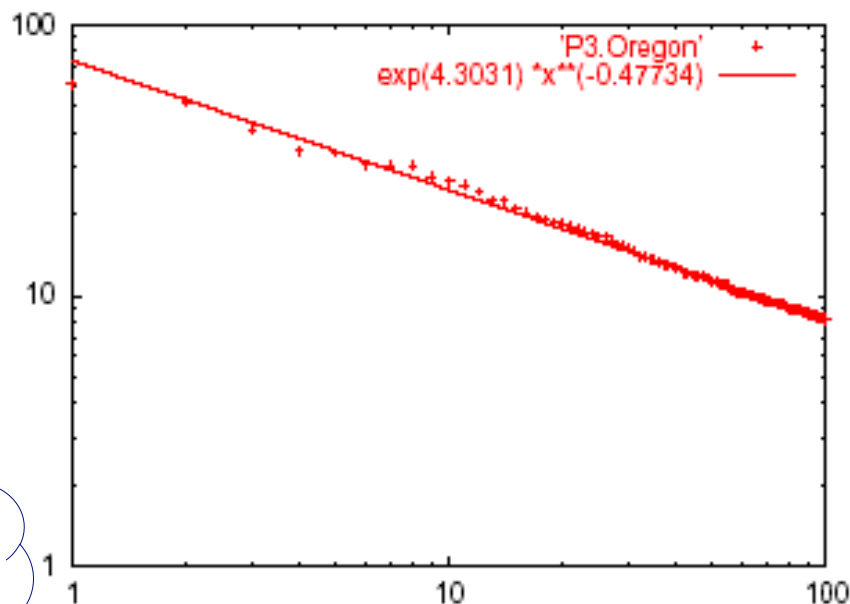
And parallelism might not help

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



Solution# S.2: Eigen Exponent E

Eigenvalue



Exponent = slope

$$E = -0.48$$

May 2001

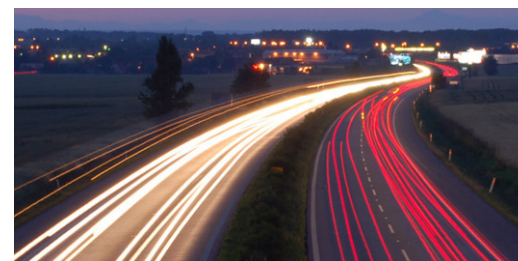
$$\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$$

Rank of decreasing eigenvalue

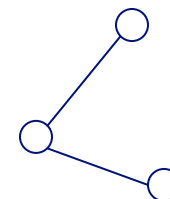
- A2: power law in the eigenvalues of the adjacency matrix ('eig()')

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
 - ➔ – Patterns: Degree; Triangles
 - Anomaly/fraud detection
 - Graph understanding
- Part#2: time-evolving graphs; tensors
- Conclusions

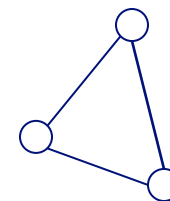


Solution# S.3: Triangle ‘Laws’

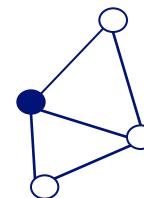


- Real social networks have a lot of triangles

Solution# S.3: Triangle ‘Laws’



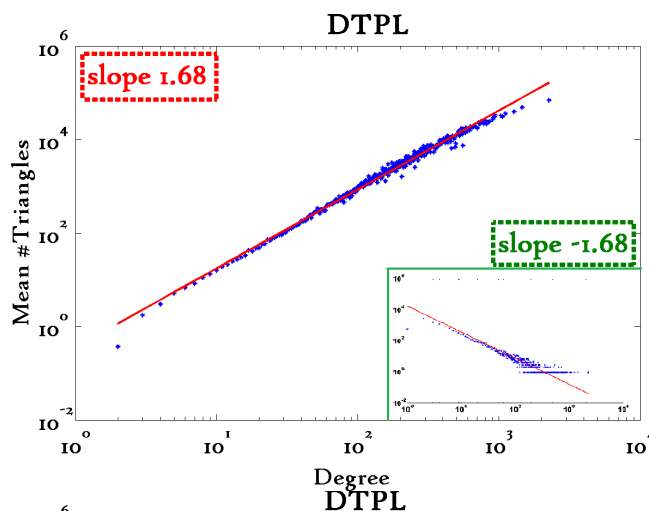
- Real social networks have a lot of triangles
 - Friends of friends are friends
- Any patterns?
 - 2x the friends, 2x the triangles ?



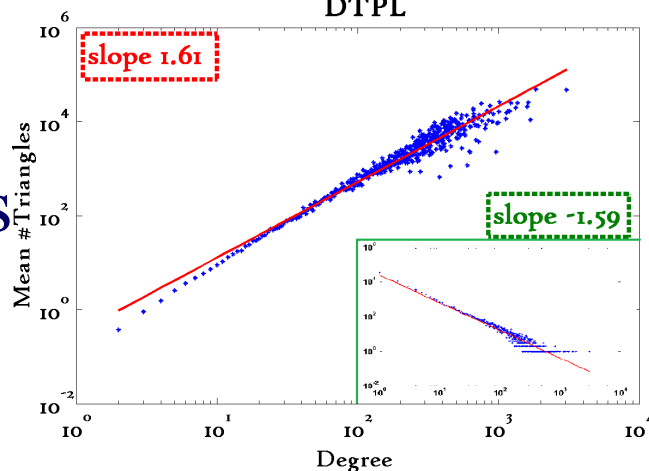
Triangle Law: #S.3

[Tsourakakis ICDM 2008]

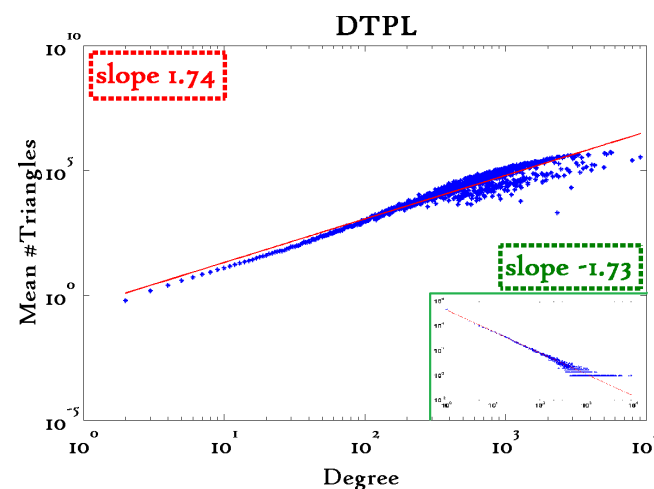
Reuters



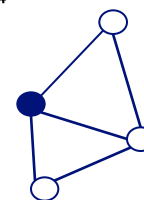
Epinions



UIC



SN



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

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:

Triangle Law: Computations

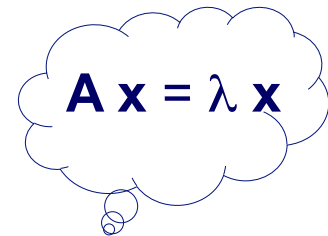
[Tsourakakis ICDM 2008]



But: triangles are expensive to compute
(3-way join; several approx. algos) – $O(d_{\max}^2)$

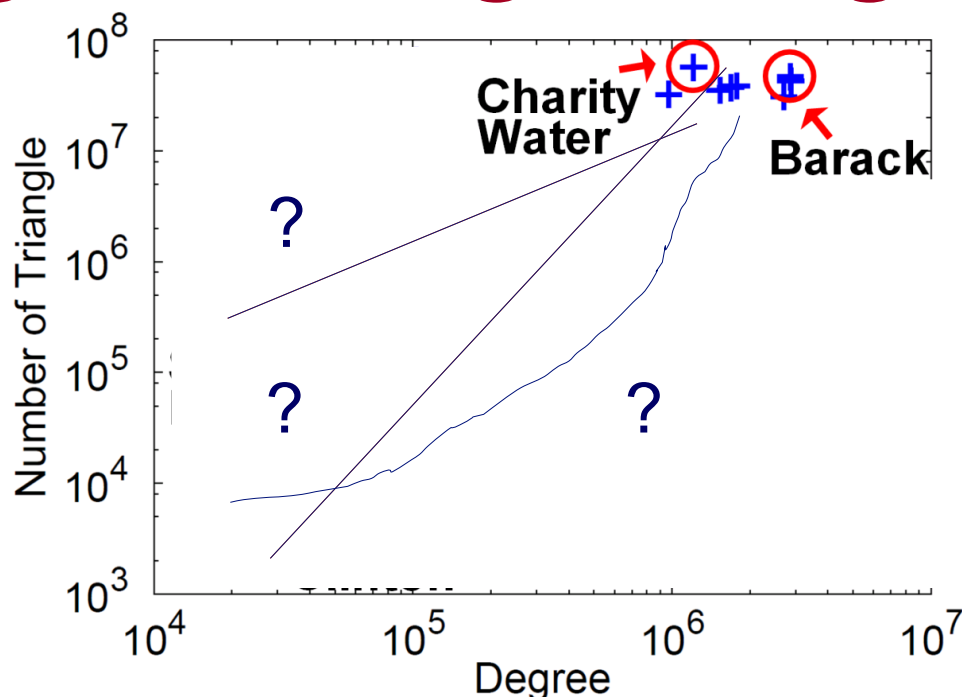
Q: Can we do that quickly?

A: Yes!



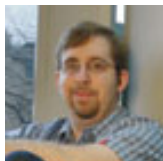
#triangles = $1/6 \text{ Sum } (\lambda_i^3)$
(and, because of skewness (S2) ,
we only need the top few eigenvalues! - $O(E)$)

Triangle counting for large graphs?

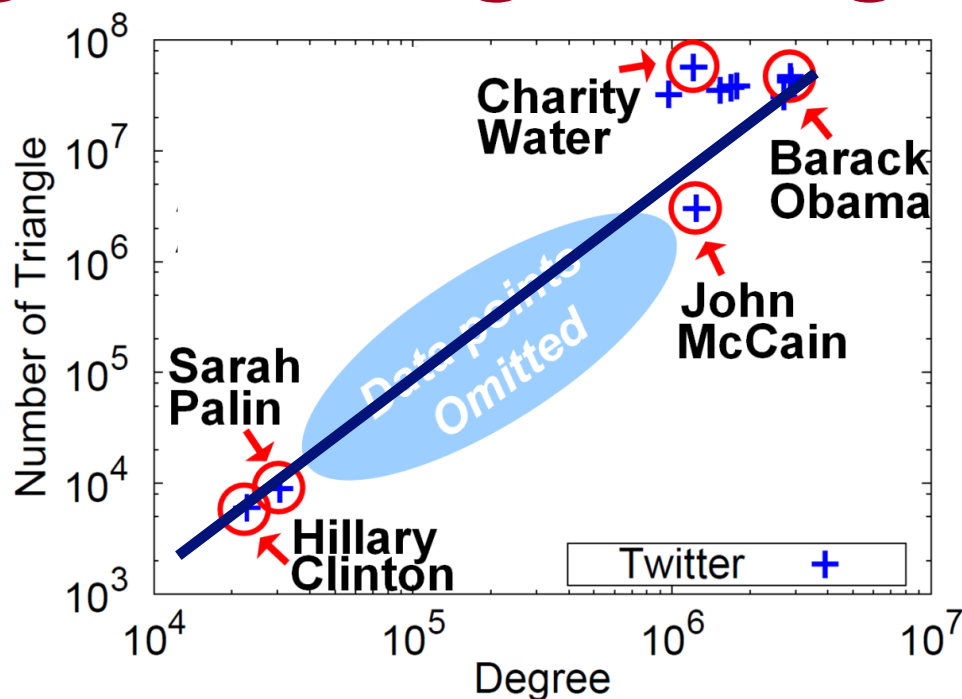


Anomalous nodes in Twitter (~ 3 billion edges)

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



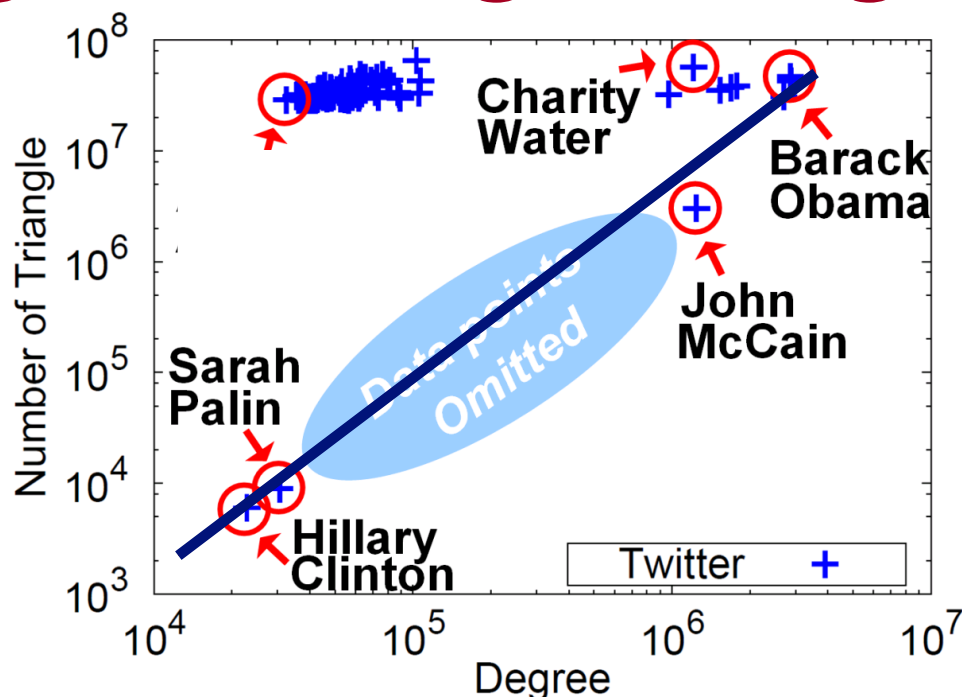
Triangle counting for large graphs?



Anomalous nodes in Twitter (~ 3 billion edges)

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

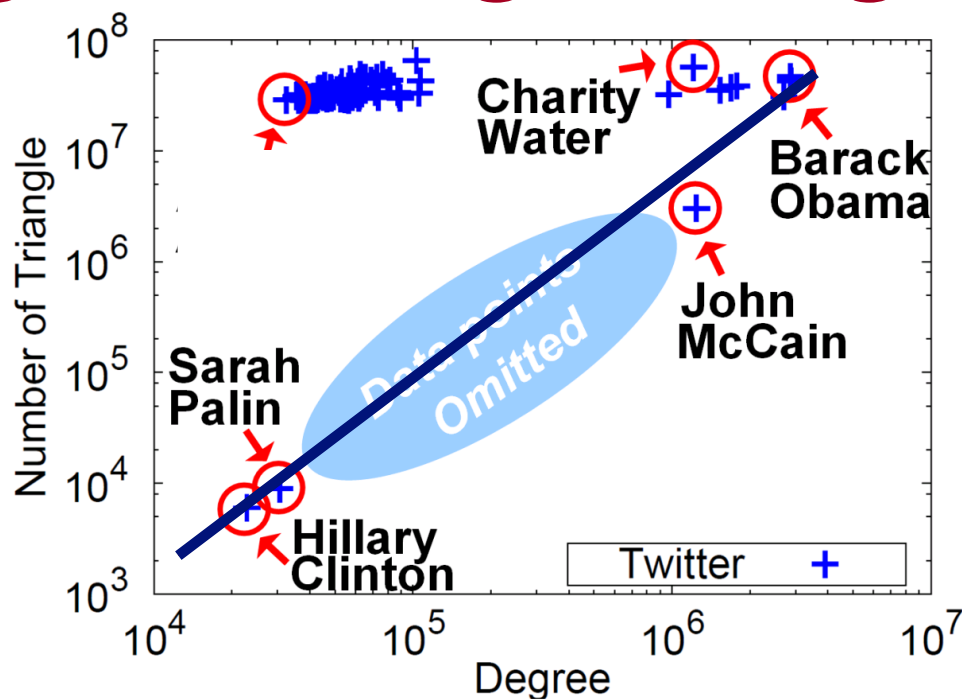
Triangle counting for large graphs?



Anomalous nodes in Twitter (~ 3 billion edges)

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

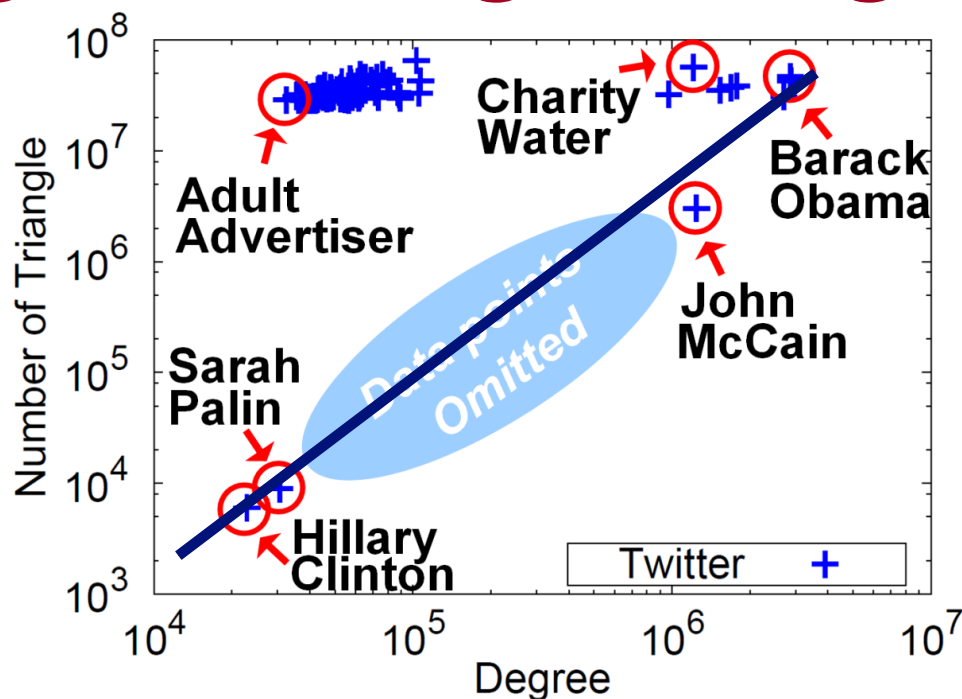
Triangle counting for large graphs?



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]

MORE Graph Patterns

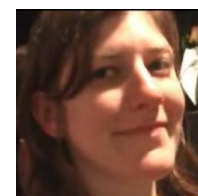
	Unweighted	Weighted
Static	 L01. Power-law degree distribution [Faloutsos et al. '99, Kleinberg et al. '99, Chakrabarti et al. '04, Newman '04]  L02. Triangle Power Law (TPL) [Tsourakakis '08]  L03. Eigenvalue Power Law (EPL) [Siganos et al. '03] L04. Community structure [Flake et al. '02, Girvan and Newman '02]	L10. Snapshot Power Law (SPL) [McGlohon et al. '08]
Dynamic	L05. Densification Power Law (DPL) [Leskovec et al. '05] L06. Small and shrinking diameter [Albert and Barabási '99, Leskovec et al. '05] L07. Constant size 2nd and 3rd connected components [McGlohon et al. '08] L08. Principal Eigenvalue Power Law (λ_1PL) [Akoglu et al. '08] 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] L02. Triangle Power Law (TPL) [Tsourakakis '08] L03. Eigenvalue Power Law (EPL) [Siganos et al. '03] 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] L06. Small and shrinking diameter [Albert and Barabási '99, Leskovec et al. '05] L07. Constant size 2 nd and 3 rd connected components [McGlohon et al. '08] L08. Principal Eigenvalue Power Law (λ_1 PL) [Akoglu et al. '08] L09. Bursty/self-similar edge/weight additions [Gomez and Stantonja '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)



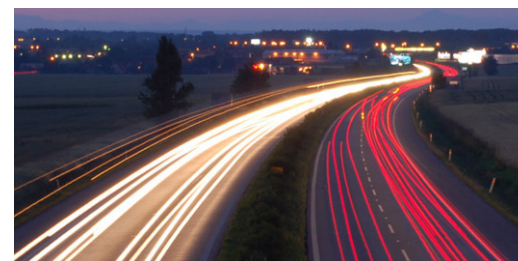
- Deepayan Chakrabarti and Christos Faloutsos, [*Graph Mining: Laws, Tools, and Case Studies*](#) Oct. 2012, Morgan Claypool.



Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs

- Patterns



- – Anomaly / fraud detection

- CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation

Patterns



anomalies

- Part#2: time-evolving graphs; tensors
- Conclusions

Fraud

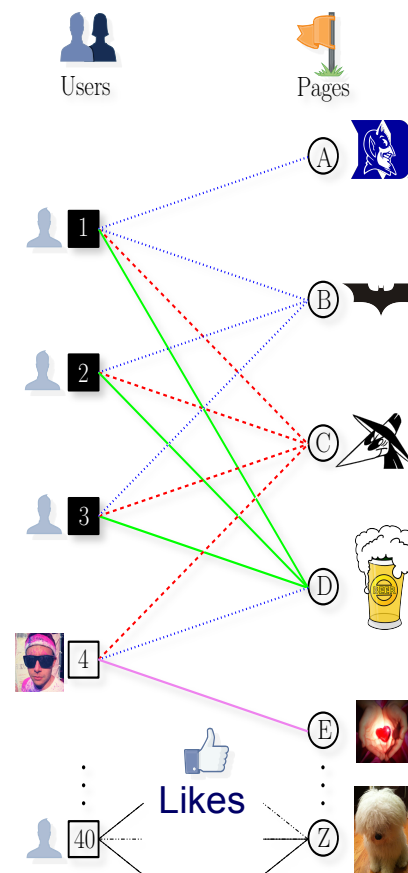
- Given
 - Who ‘likes’ what page, and when
- Find
 - Suspicious users and suspicious products



CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks, Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, Christos Faloutsos *WWW*, 2013.

Fraud

- Given
 - Who ‘likes’ what page, and when
- Find
 - Suspicious users and suspicious products



CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks, Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, Christos Faloutsos *WWW, 2013*.

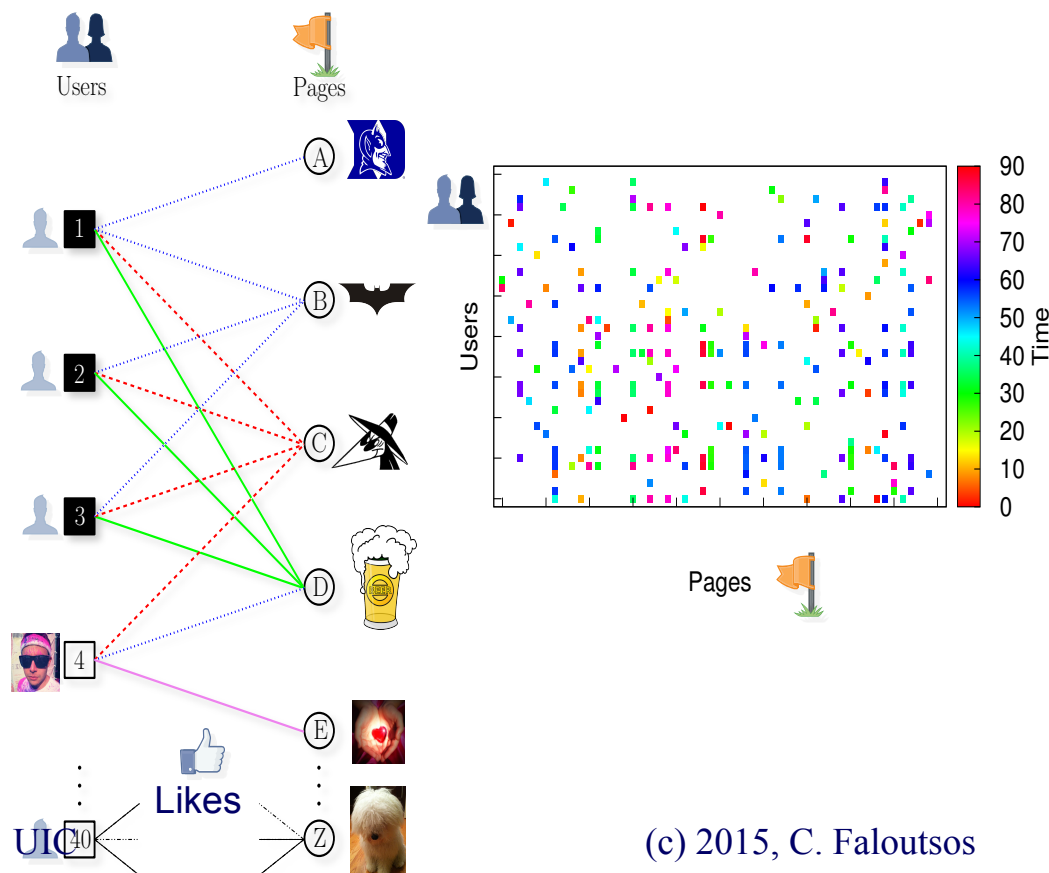
Graph Patterns and Lockstep Behavior

Our intuition

Behavior



- Lockstep behavior: Same Likes, same time



(c) 2015, C. Faloutsos

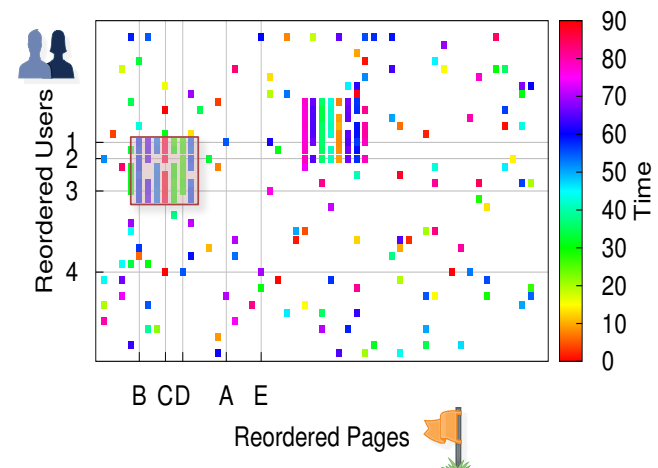
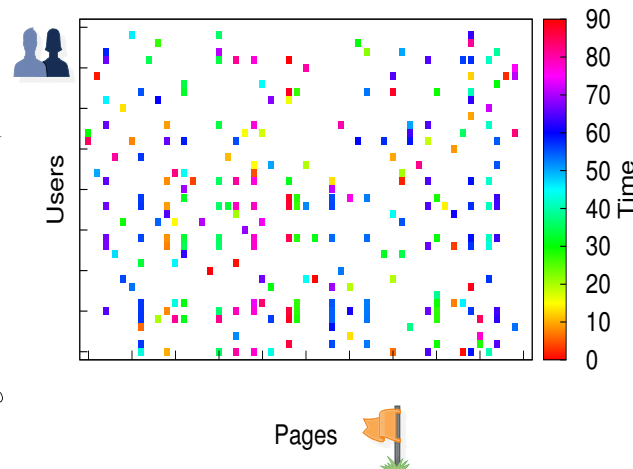
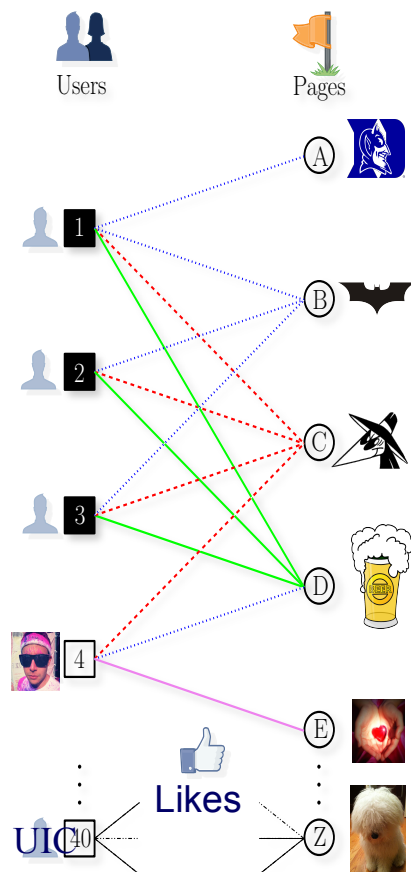


Graph Patterns and Lockstep Behavior

Our intuition



- Lockstep behavior: Same Likes, same time



December 2013

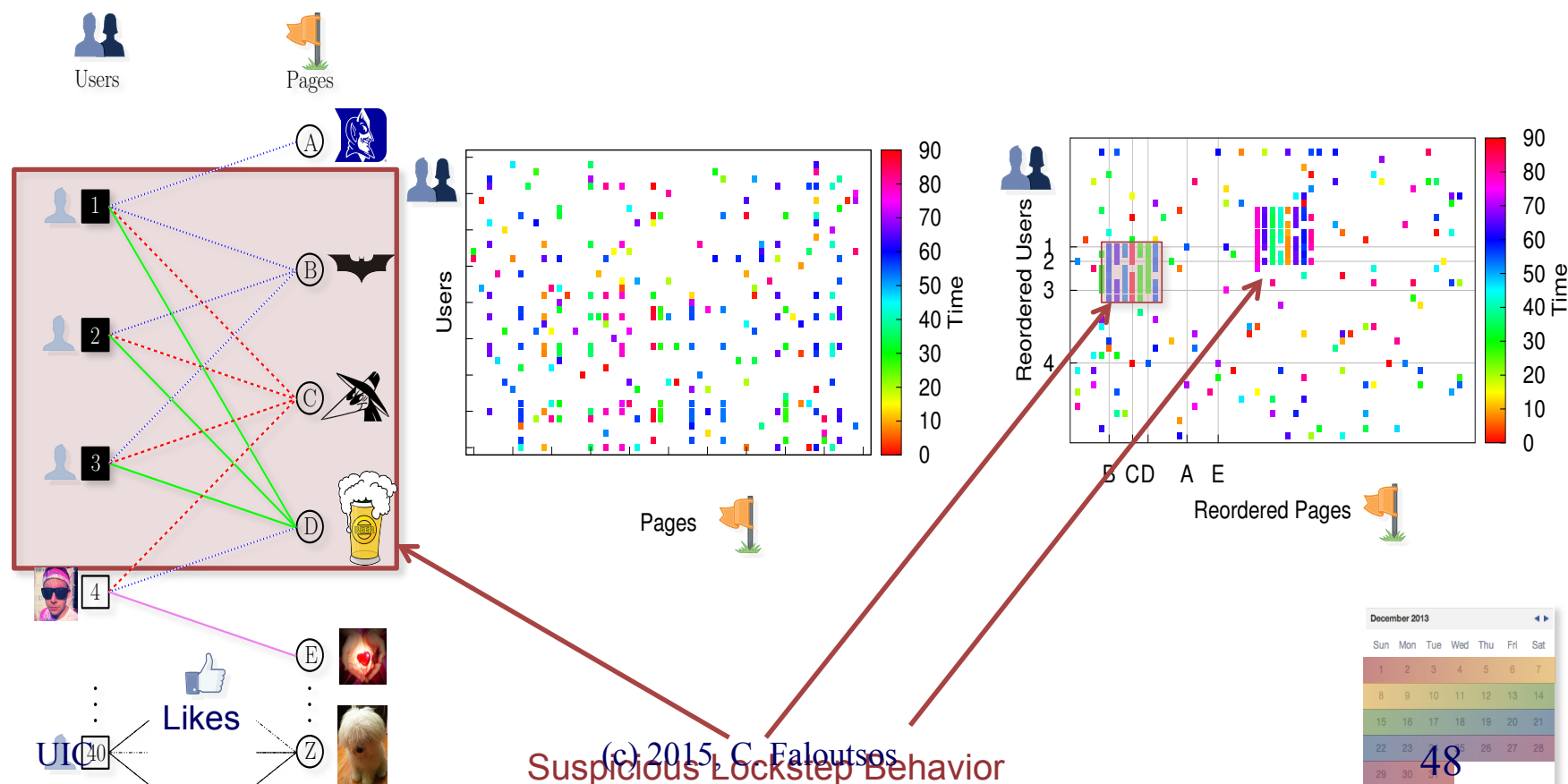
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Graph Patterns and Lockstep Behavior

Our intuition



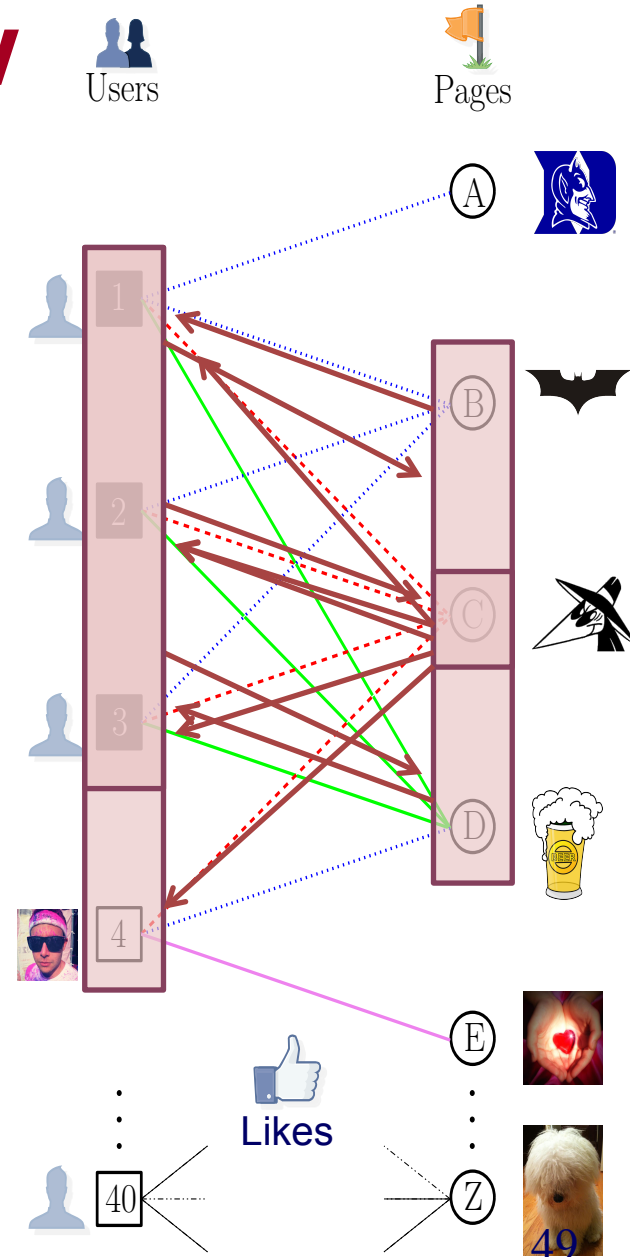
- Lockstep behavior: Same Likes, same time



Suspicious Lockstep Behavior

MapReduce Overview

- Use Hadoop to search for many clusters in parallel:
 - Start with randomly seed
 - Update set of Pages and center Like times for each cluster
 - Repeat until convergence

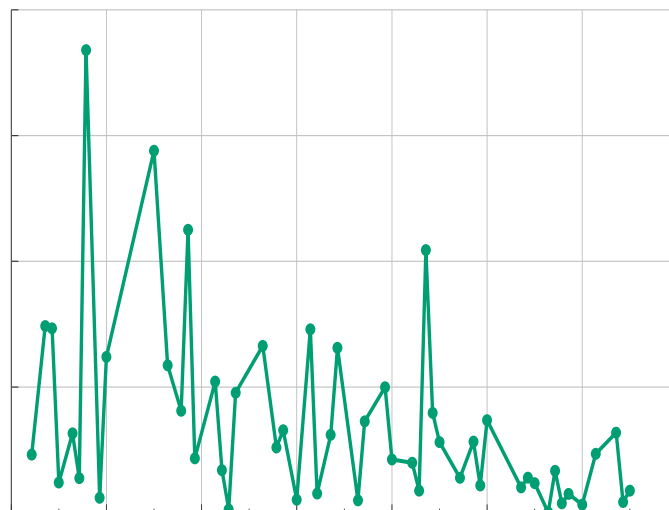


Deployment at Facebook

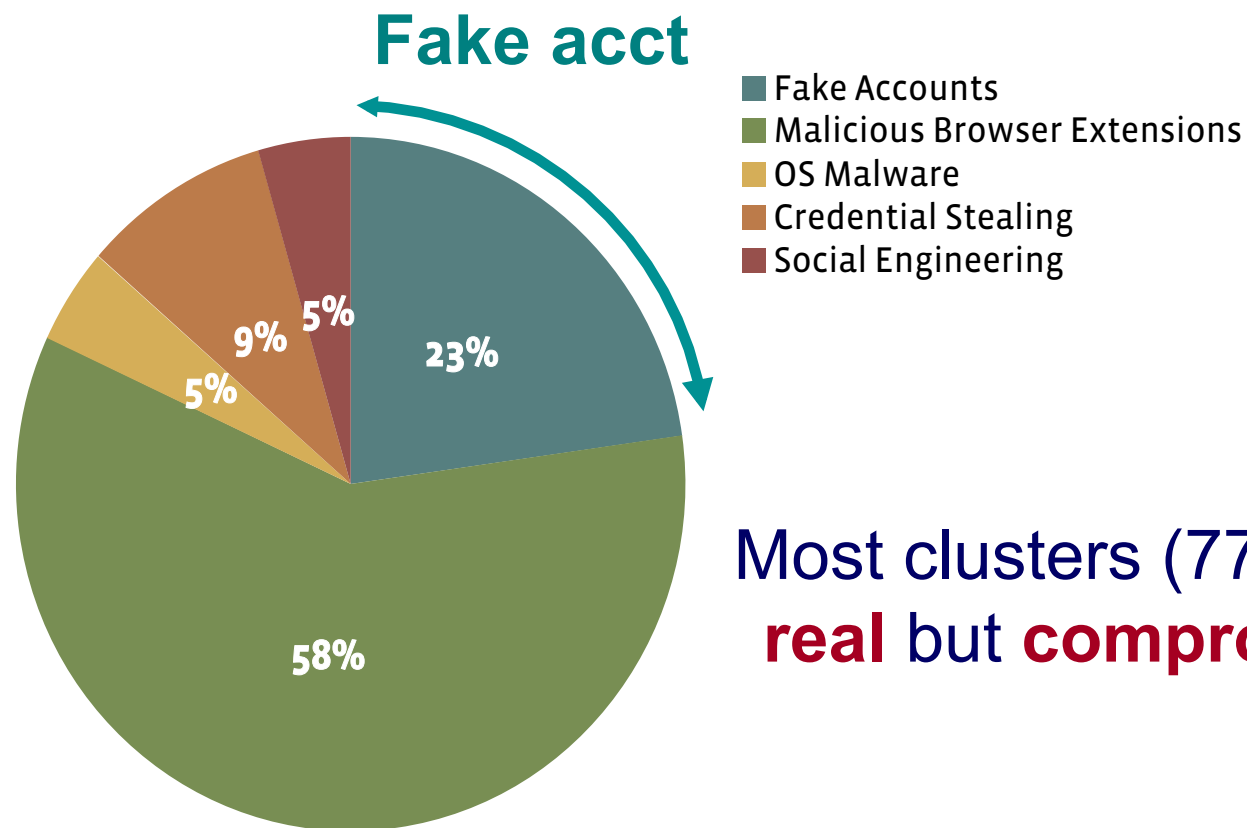
- *CopyCatch* runs regularly (along with many other security mechanisms, and a large Site Integrity team)

3 months of *CopyCatch* @ Facebook

#users
caught



Deployment at Facebook

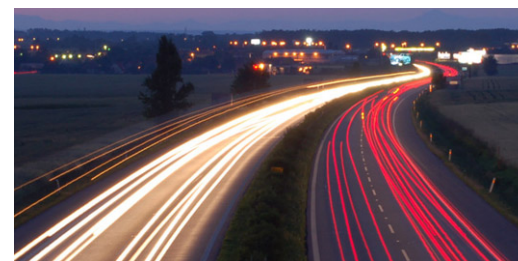


Most clusters (77%) come from **real** but **compromised** users

Manually labeled 22 randomly selected *clusters* from February 2013

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
 - Patterns
 - Anomaly / fraud detection
 - CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions



Problem: Social Network Link Fraud

Target: find “stealthy” attackers missed by other algorithms

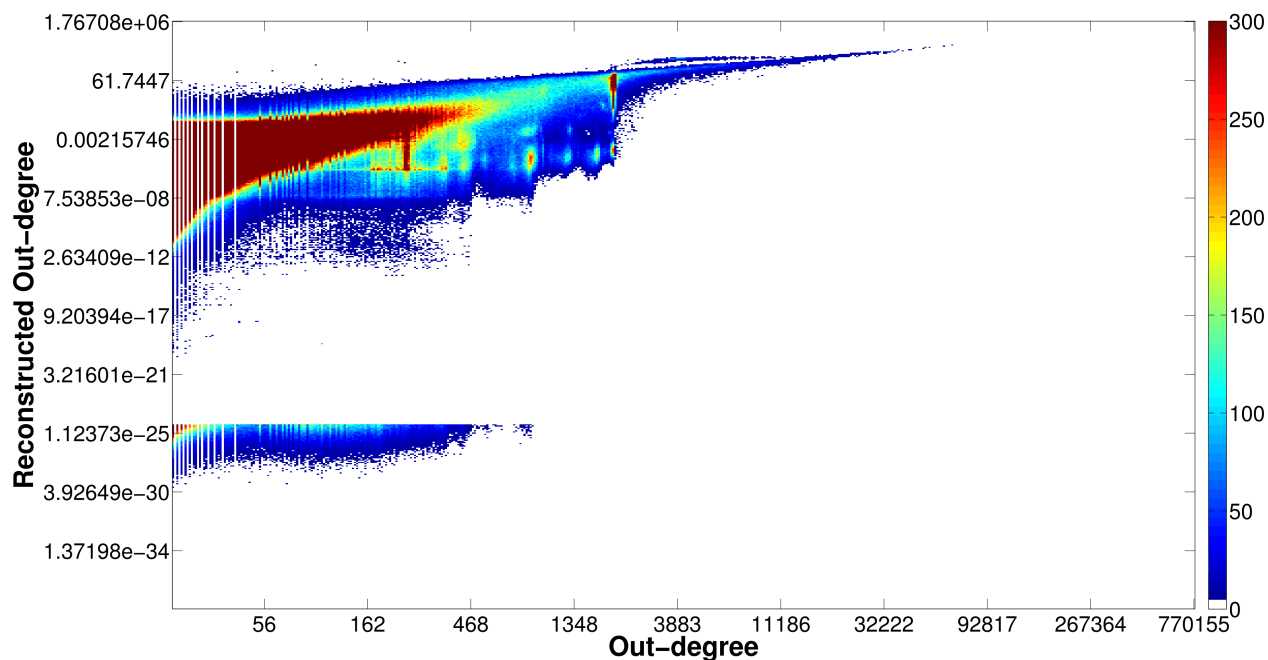


Clique

41.7M nodes
1.5B edges



Bipartite
core



Problem: Social Network Link Fraud

Target: find “stealthy” attackers missed by other algorithms



Lekan Olawole Lowe @loweinc

26 Jul 09

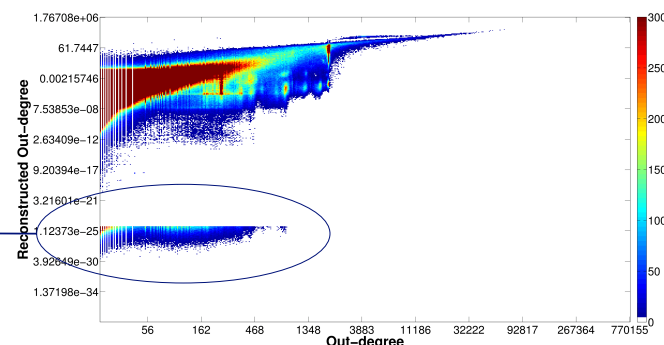
Sign up free and Get 400 followers a day using <http://tweeteradder.com>



Lekan Olawole Lowe @loweinc

26 Jul 09

Get 400 followers a day using <http://www.tweeterfollow.com>



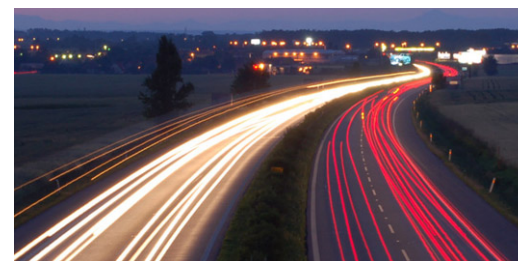
Takeaway: use *reconstruction error* between true/latent representation!



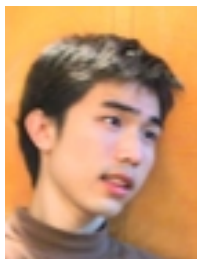
Neil Shah, Alex Beutel, Brian Gallagher and Christos Faloutsos. *Spotting Suspicious Link Behavior with fBox: An Adversarial Perspective*. ICDM 2014, Shenzhen, China.

Roadmap

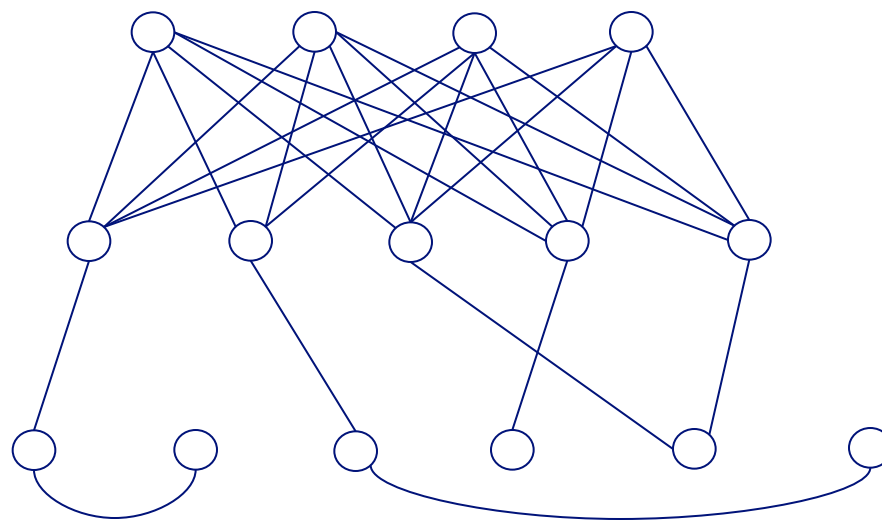
- Introduction – Motivation
- Part#1: Patterns in graphs
 - Patterns
 - Anomaly / fraud detection
 - CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions



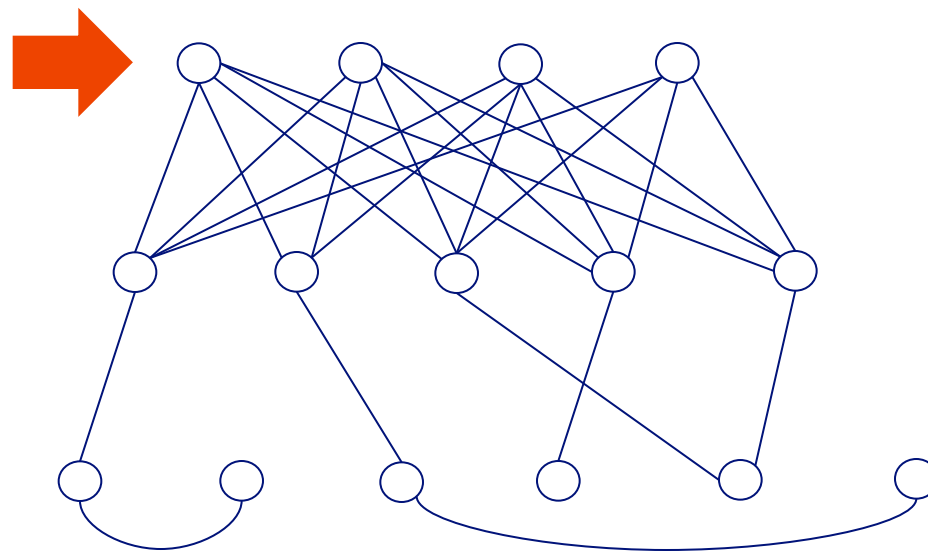
E-bay Fraud detection



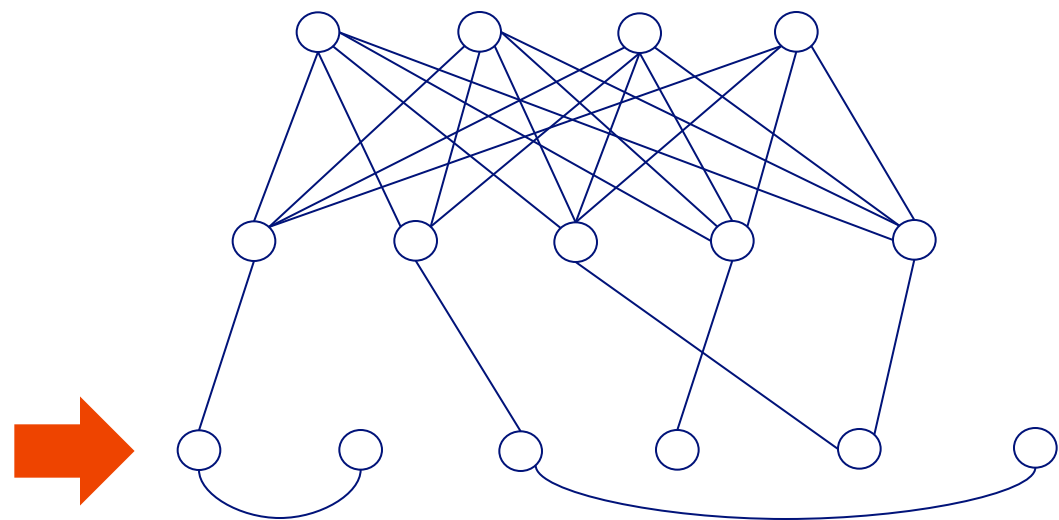
w/ Polo Chau &
Shashank Pandit, CMU
[www'07]



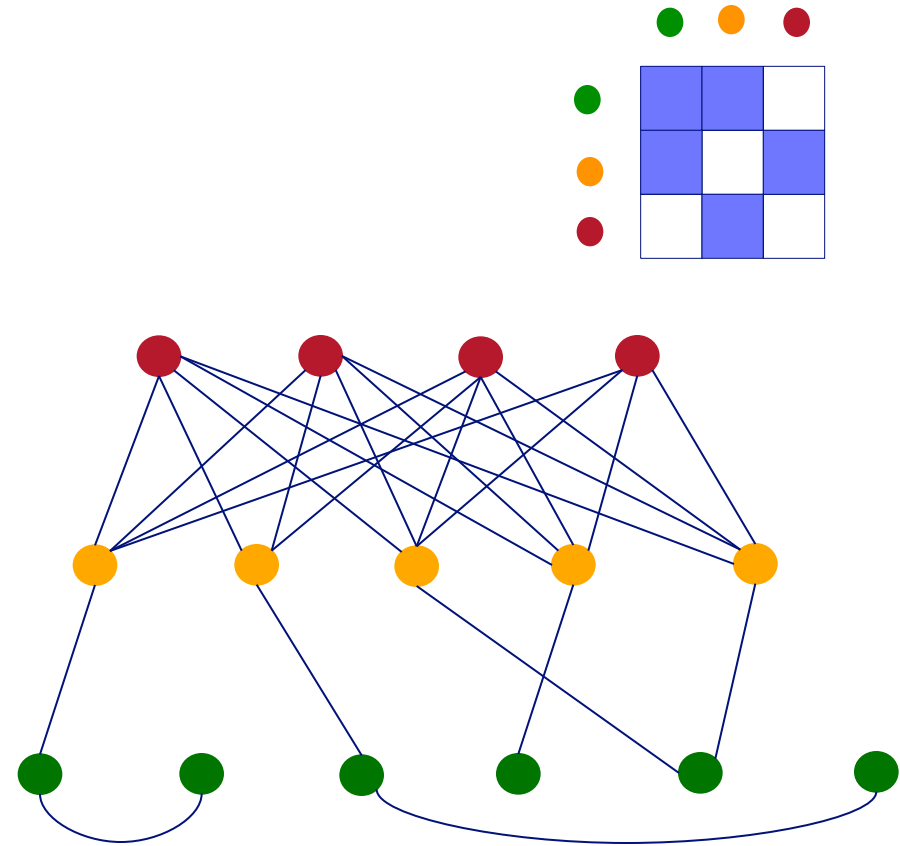
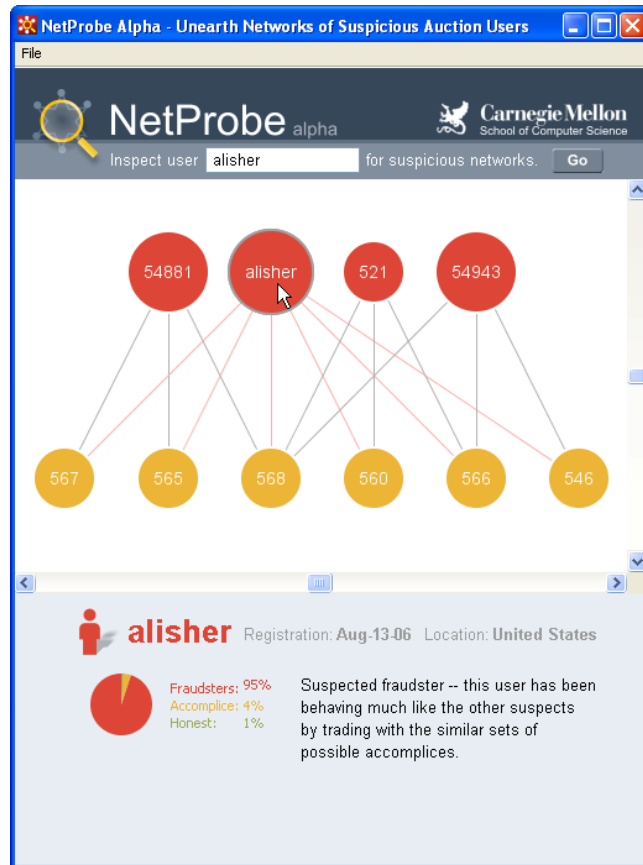
E-bay Fraud detection



E-bay Fraud detection



E-bay Fraud detection - NetProbe



Popular press



The Washington Post

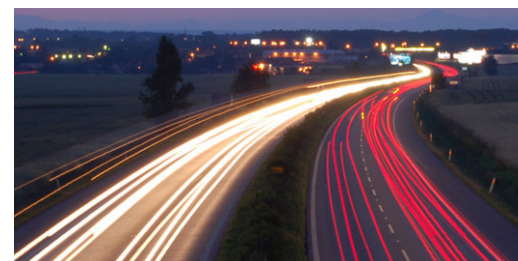
Los Angeles Times

And less desirable attention:

- E-mail from ‘Belgium police’ (‘copy of your code?’)

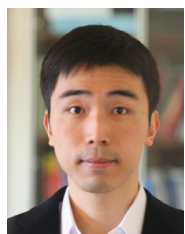
Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
 - Patterns
 - Anomaly / fraud detection
 - CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation; antivirus app
- Part#2: time-evolving graphs; tensors
- Conclusions



Polonium: Tera-Scale Graph Mining and Inference for Malware Detection

SDM 2011, Mesa, Arizona



Polo Chau

Machine Learning Dept



Carey Nachenberg

Vice President & Fellow



Jeffrey Wilhelm

Principal Software Engineer



Adam Wright

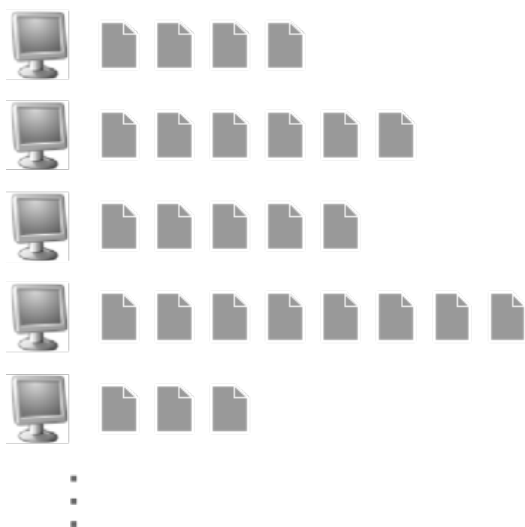
Software Engineer



Prof. Christos Faloutsos

Computer Science Dept

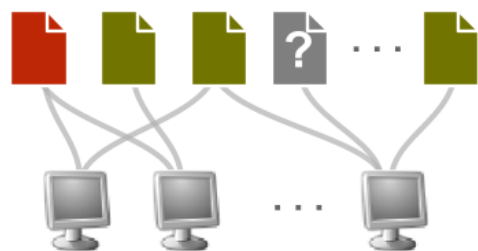
Polonium: The Data



60+ terabytes of data *anonymously*
contributed by participants of worldwide
Norton Community Watch program

50+ million machines

900+ million executable files



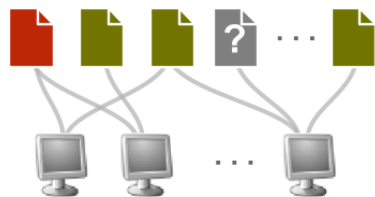
Constructed a machine-file bipartite graph
(0.2 TB+)

1 billion nodes (machines and files)

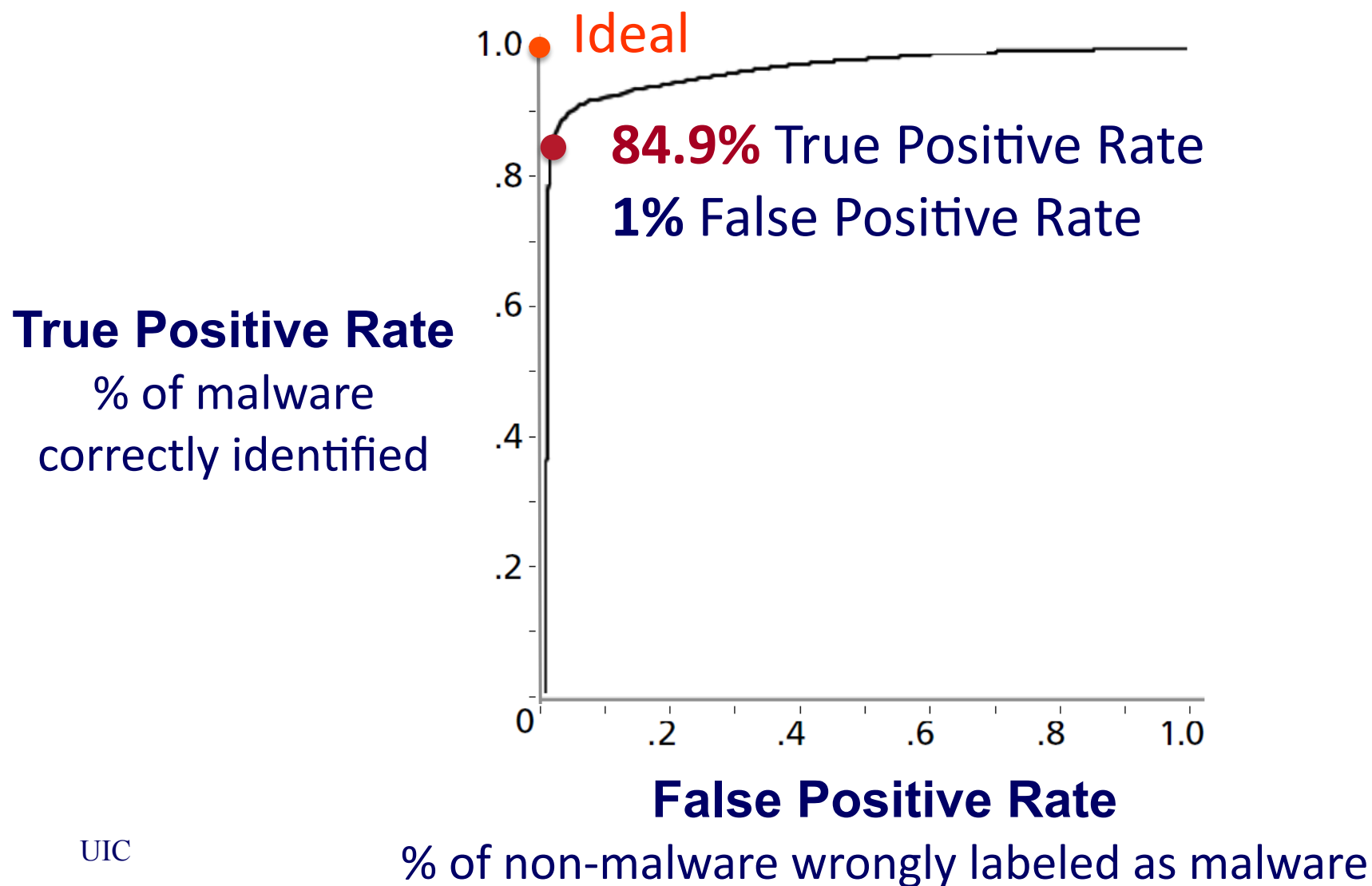
37 billion edges

Polonium: Key Ideas

- Use **Belief Propagation** to propagate domain knowledge in machine-file graph to detect malware
- Use “**guilt-by-association**” (i.e., homophily)
 - E.g., files that appear on machines with many bad files are more likely to be bad
- Scalability: handles 37 billion-edge graph

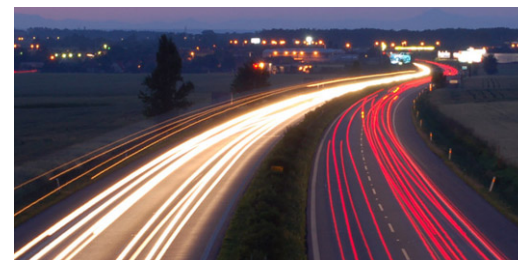


Polonium: One-Interaction Results



Roadmap

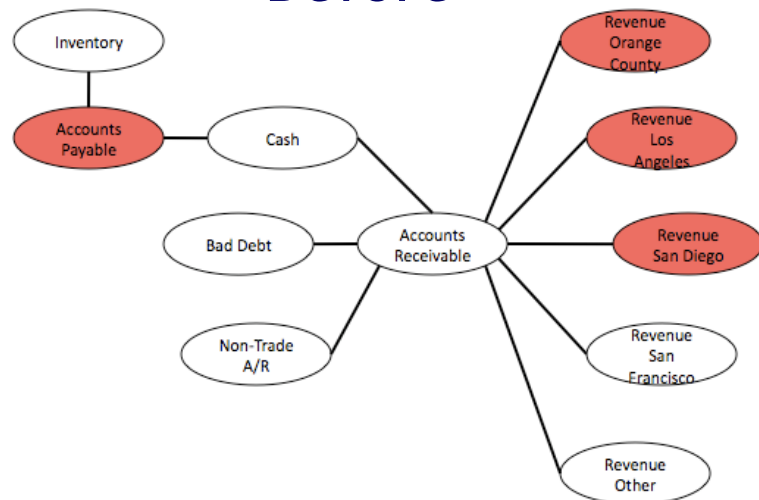
- Introduction – Motivation
- Part#1: Patterns in graphs
 - Patterns
 - Anomaly / fraud detection
 - CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation; financial fraud
- Part#2: time-evolving graphs; tensors
- Conclusions



Network Effect Tools: SNARE

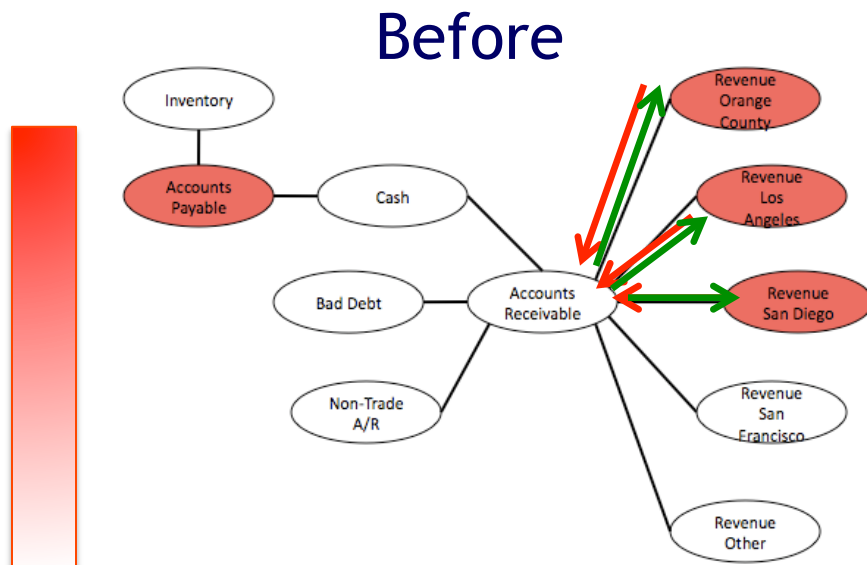
- Some accounts are sort-of-suspicious – how to combine weak signals?

Before



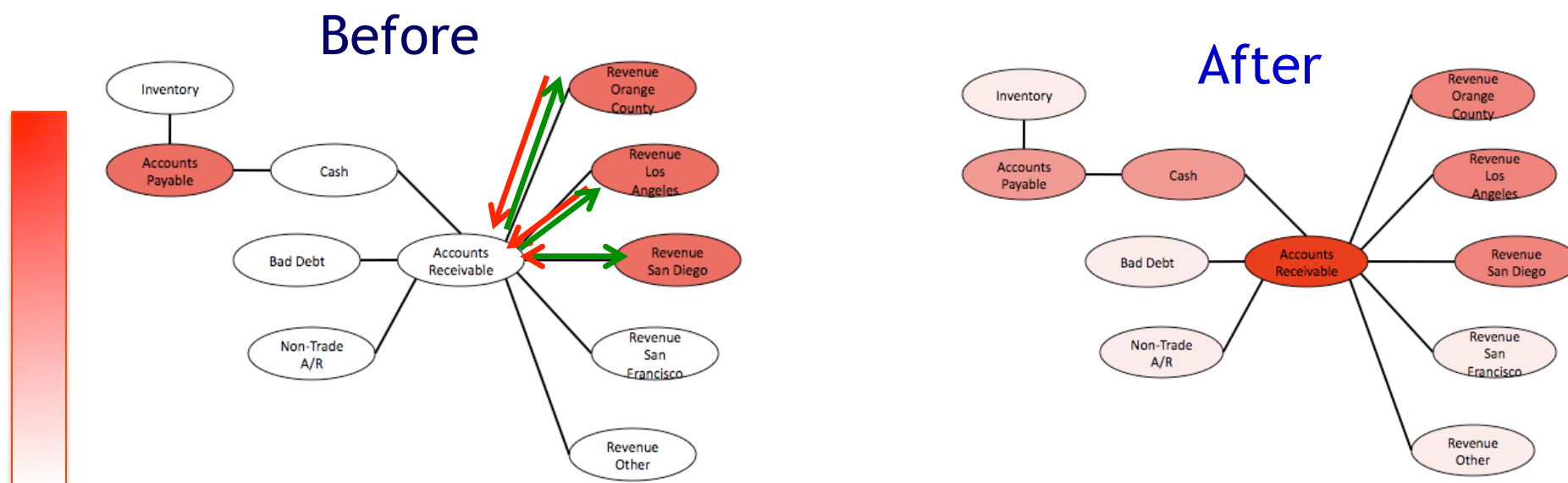
Network Effect Tools: SNARE

- A: Belief Propagation.



Network Effect Tools: SNARE

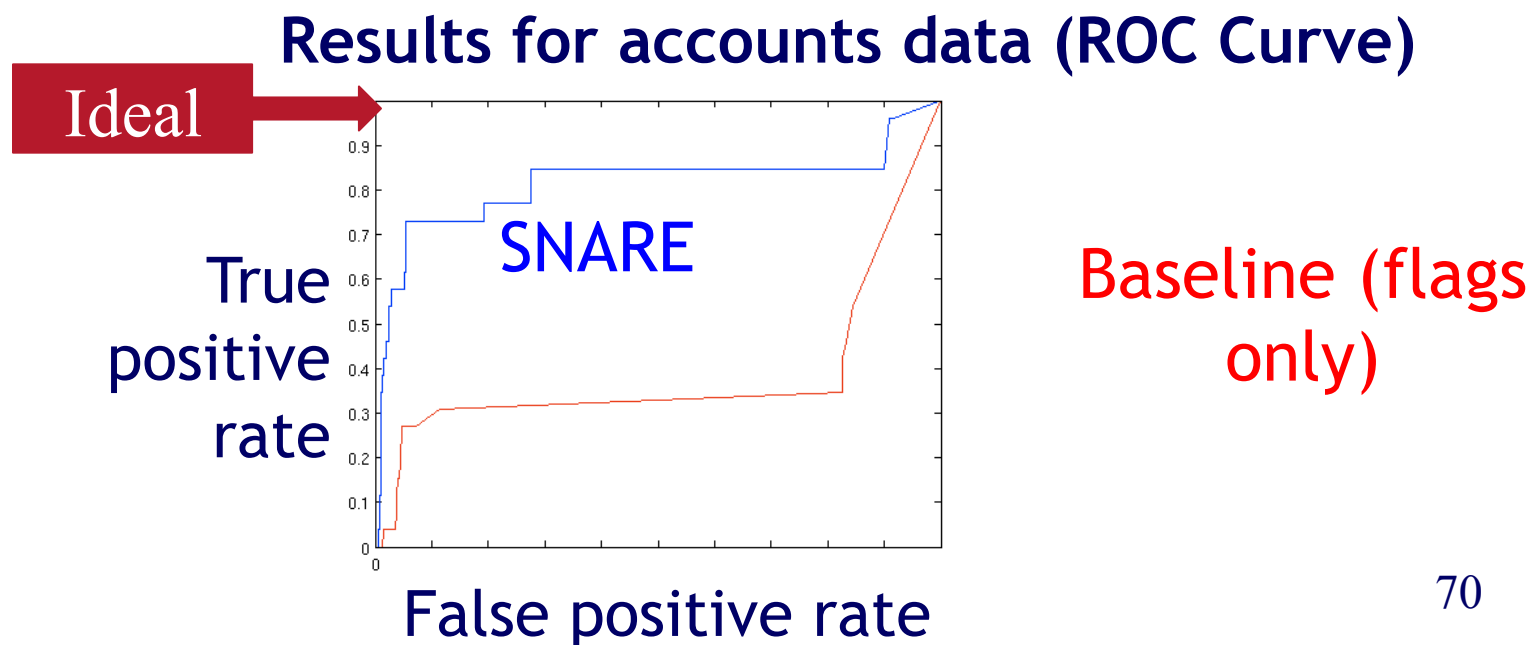
- A: Belief Propagation.



Mary McGlohon, Stephen Bay, Markus G. Anderle, David M. Steier, Christos Faloutsos: *SNARE: a link analytic system for graph labeling and risk detection*. KDD 2009: 1265-1274

Network Effect Tools: SNARE

- Produces improvement over simply using flags
 - Up to 6.5 lift
 - Improvement especially for low false positive rate

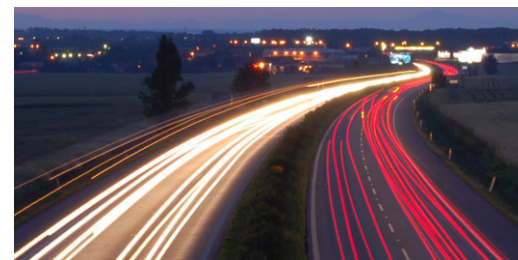


Network Effect Tools: SNARE

- **Accurate**- Produces large improvement over simply using flags
- **Flexible**- Can be applied to other domains
- **Scalable**- One iteration BP runs in linear time (# edges)
- **Robust**- Works on large range of parameters

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
 - Patterns
 - Anomaly / fraud detection
 - CopyCatch
 - Spectral methods ('fBox')
 - Belief Propagation; fast computation & unification
- Part#2: time-evolving graphs; tensors
- Conclusions



Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms



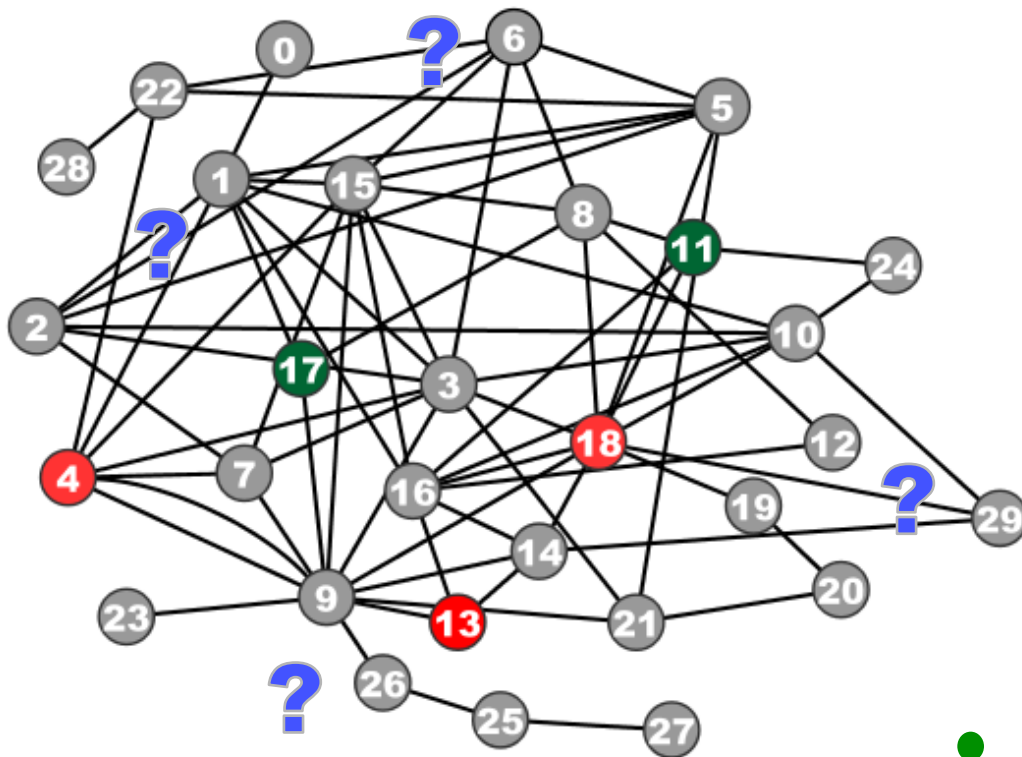
Danai Koutra
U Kang
Hsing-Kuo Kenneth Pao

Tai-You Ke
Duen Horng (Polo) Chau
Christos Faloutsos

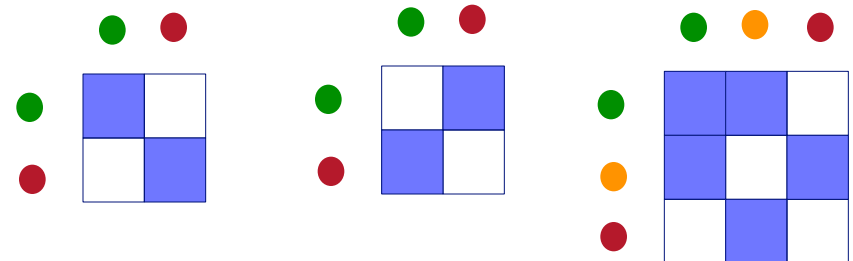
ECML PKDD, 5-9 September 2011, Athens, Greece

Problem Definition:

GBA techniques



Given: Graph; &
few labeled nodes
Find: labels of rest
(assuming network
effects)



Are they related?

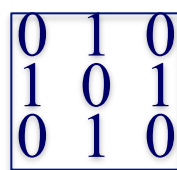
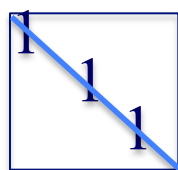
- RWR (Random Walk with Restarts)
 - google's pageRank (*'if my friends are important, I'm important, too'*)
- SSL (Semi-supervised learning)
 - minimize the differences among neighbors
- BP (Belief propagation)
 - send messages to neighbors, on what you believe about them

Are they related? **YES!**

- RWR (Random Walk with Restarts)
 - google's pageRank (*'if my friends are important, I'm important, too'*)
- SSL (Semi-supervised learning)
 - minimize the differences among neighbors
- BP (Belief propagation)
 - send messages to neighbors, on what you believe about them

Correspondence of Methods

Method	Matrix	Unknown	known
RWR	$[I - c \underline{AD}^{-1}]$	$\times$	$\mathbf{x} = (1-c)\mathbf{y}$
SSL	$[I + a(\mathbf{D} - \underline{A})]$	$\times$	$\mathbf{x} = \mathbf{y}$
FABP	$[I + a \mathbf{D} - c' \underline{A}]$	$\times$	$\mathbf{b}_h = \phi_h$



adjacency
matrix

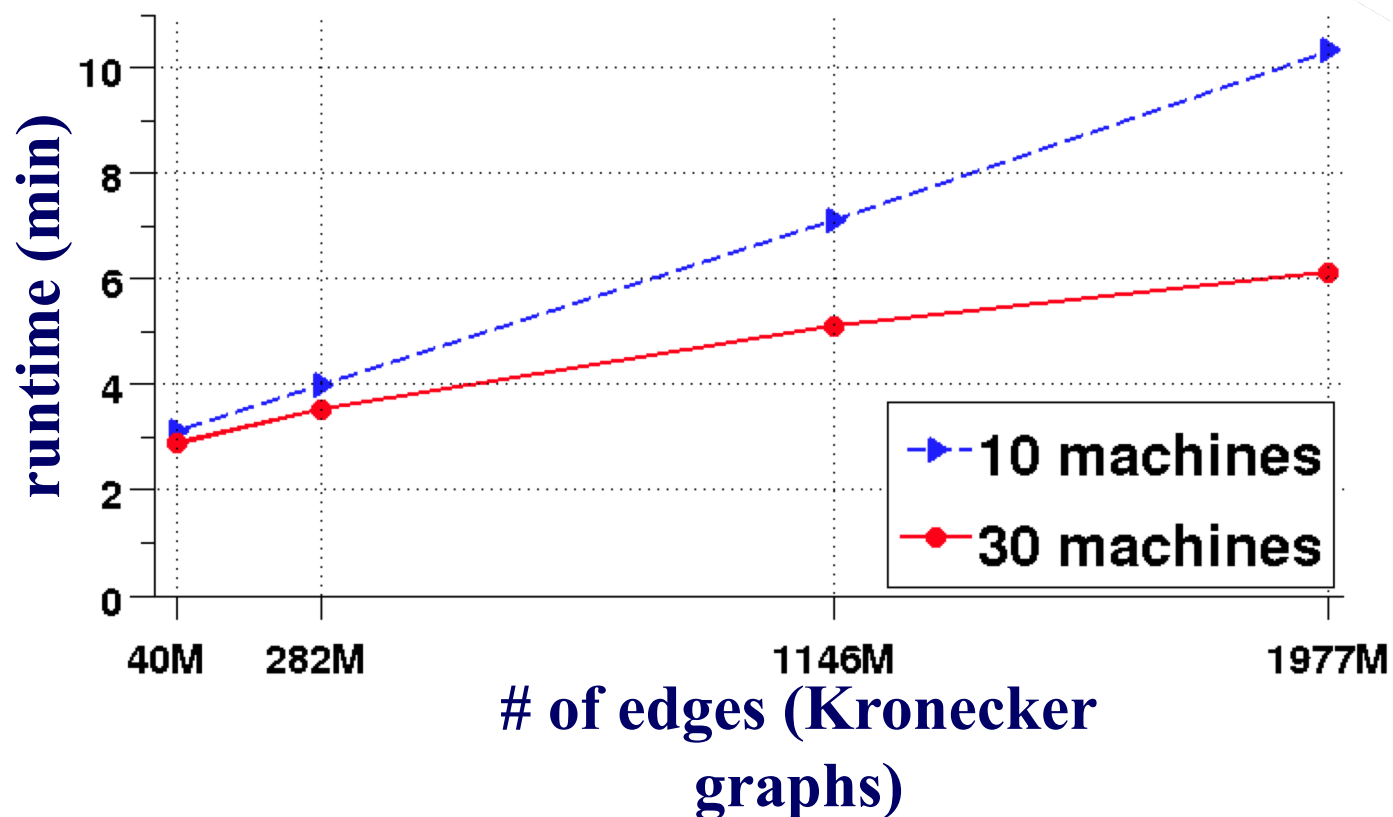
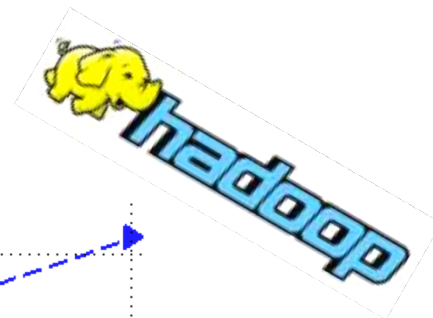


final
labels/
beliefs

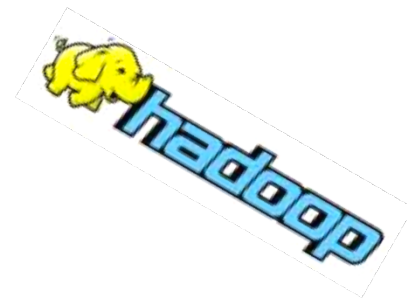


prior
labels/
beliefs

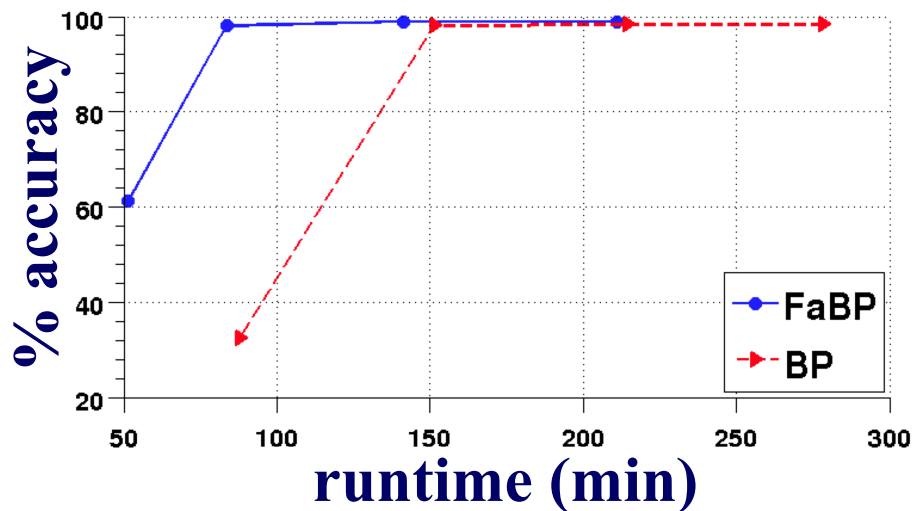
Results: Scalability



FABP is **linear** on the number of edges.



Results: Parallelism



**FABP ~2x faster
& wins/ties on accuracy.**

Summary of Part#1

- *many* patterns in real graphs
 - Power-laws everywhere
 - Gaussian trap
 - $Avg \ll Max$
 - Long (and growing) list of tools for anomaly/fraud detection



Patterns  anomalies

Roadmap

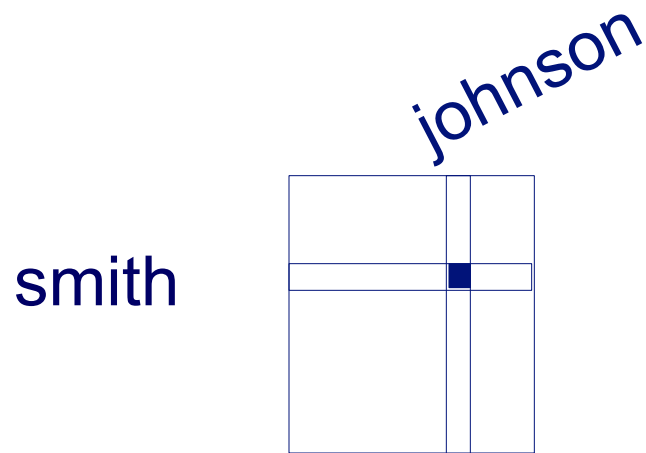
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
 - ➔ – P2.1: time-evolving graphs
 - P2.2: with side information (‘coupled’ M.T.F.)
 - Speed
- Part#3: Cascades and immunization
- Conclusions



Part 2: Time evolving graphs; tensors

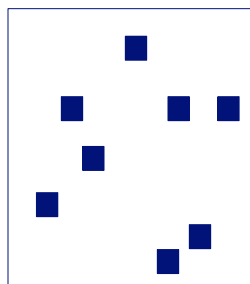
Graphs over time -> tensors!

- Problem #2.1:
 - Given who calls whom, and when
 - Find patterns / anomalies



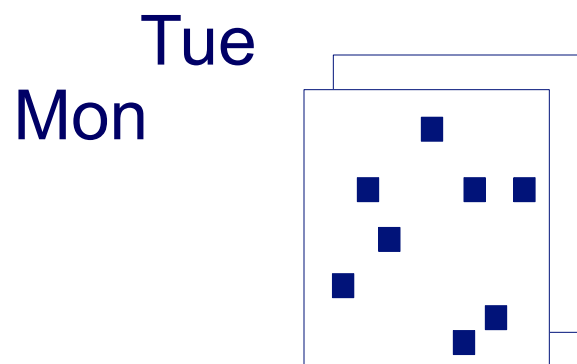
Graphs over time -> tensors!

- Problem #2.1:
 - Given who calls whom, and when
 - Find patterns / anomalies



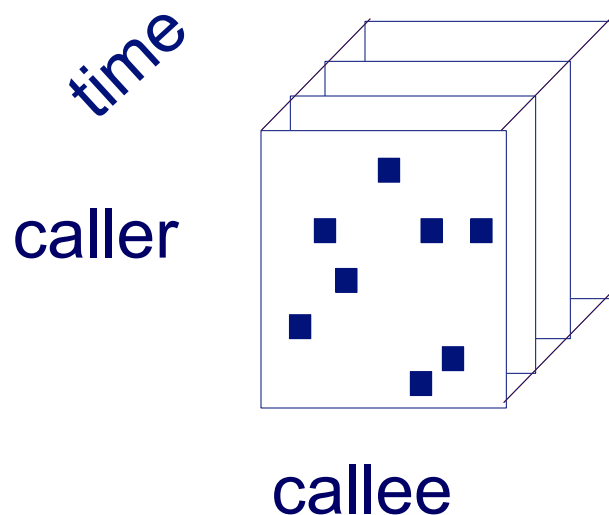
Graphs over time -> tensors!

- Problem #2.1:
 - Given who calls whom, and when
 - Find patterns / anomalies



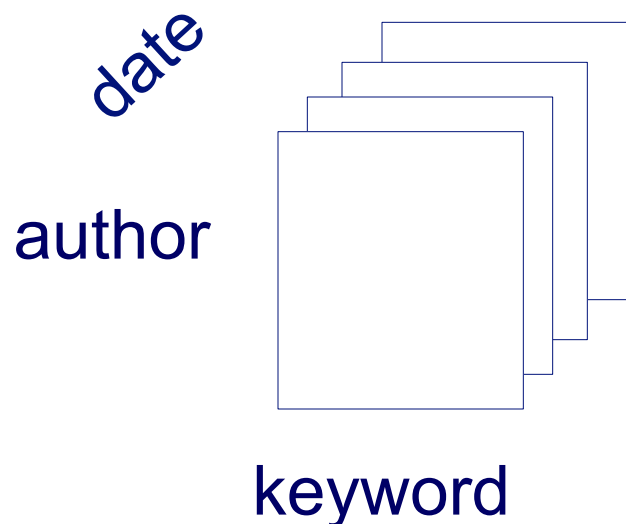
Graphs over time -> tensors!

- Problem #2.1:
 - Given who calls whom, and when
 - Find patterns / anomalies



Graphs over time -> tensors!

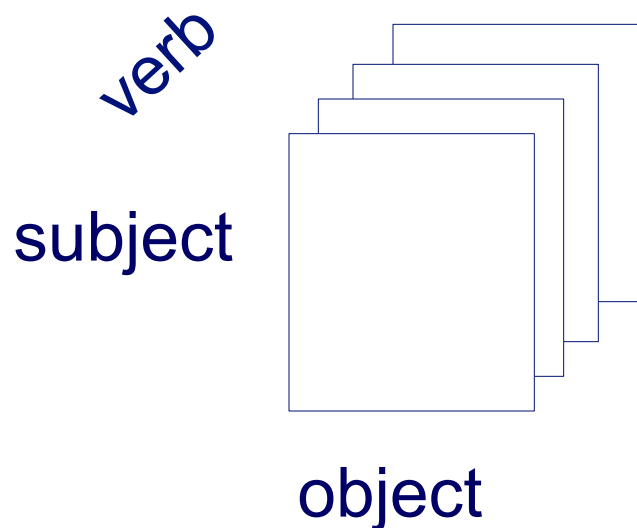
- Problem #2.1':
 - Given author-keyword-date
 - Find patterns / anomalies



MANY more settings,
with >2 'modes'

Graphs over time -> tensors!

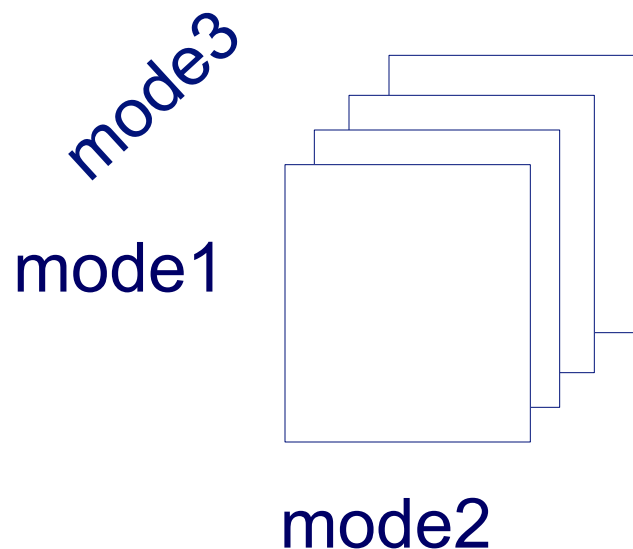
- Problem #2.1’’:
 - Given subject – verb – object facts
 - Find patterns / anomalies



MANY more settings,
with >2 ‘modes’

Graphs over time -> tensors!

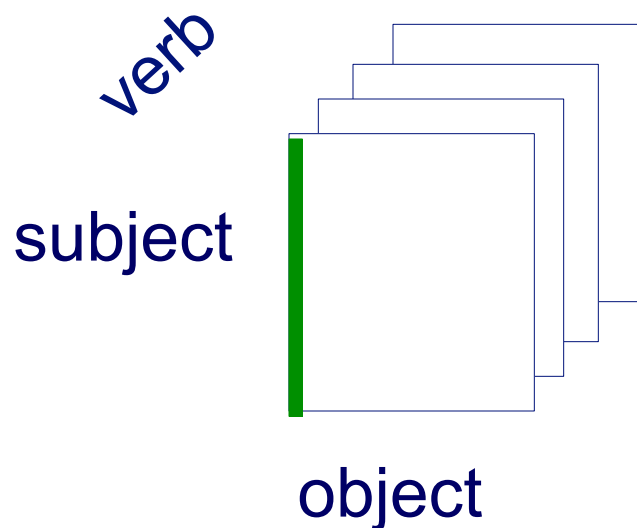
- Problem #2.1''':
 - Given <triplets>
 - Find patterns / anomalies



MANY more settings,
with >2 'modes'
(and 4, 5, etc modes)

Graphs & side info

- Problem #2.2: coupled (eg., side info)
 - Given subject – verb – object facts
 - And voxel-activity for each subject-word
 - Find patterns / anomalies

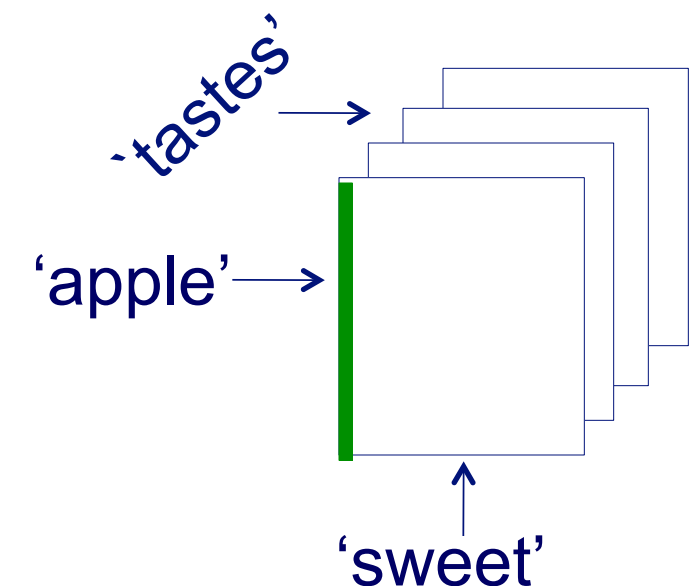


fMRI voxel activity

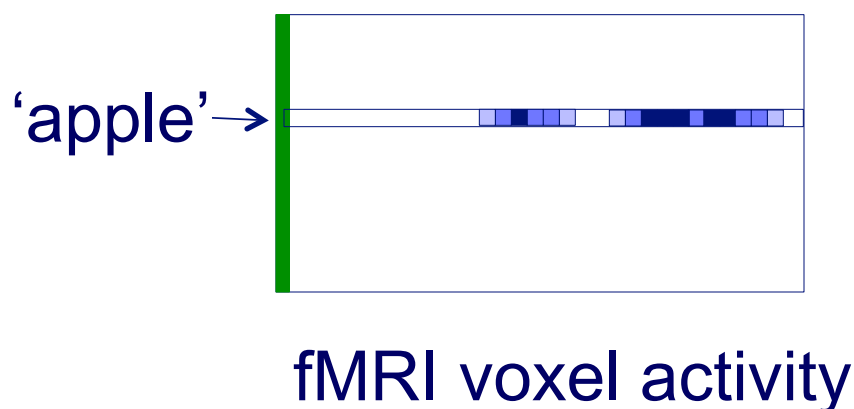
`apple tastes sweet'

Graphs & side info

- Problem #2.2: coupled (eg., side info)
 - Given subject – verb – object facts
 - And voxel-activity for each subject-word
 - Find patterns / anomalies



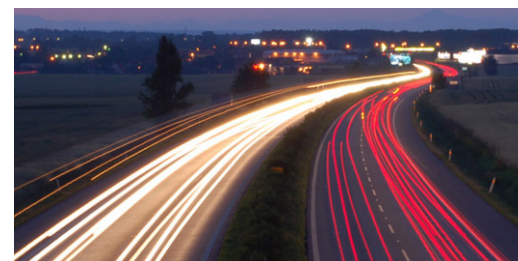
'apple tastes sweet'



fMRI voxel activity

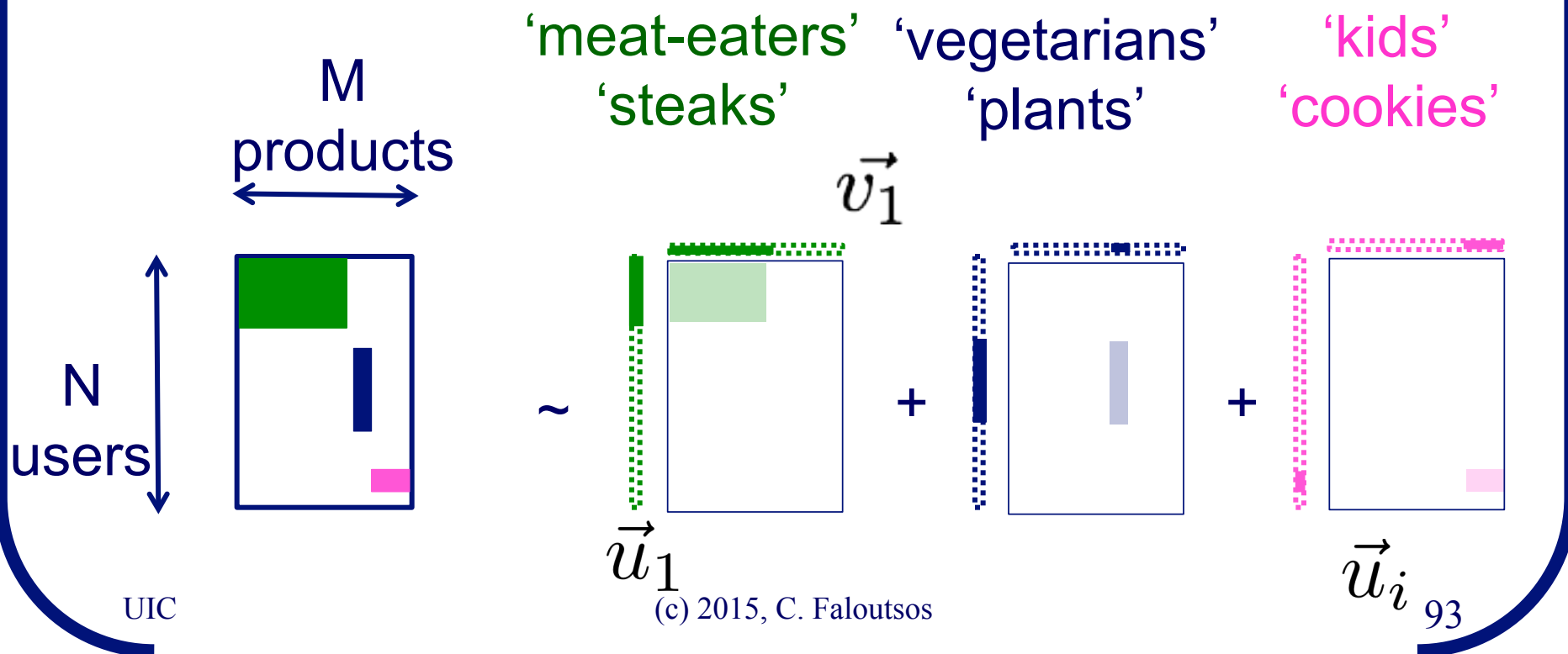
Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
 - ➔ – P2.1: time-evolving graphs
 - P2.2: with side information (‘coupled’ M.T.F.)
 - Speed
- Conclusions



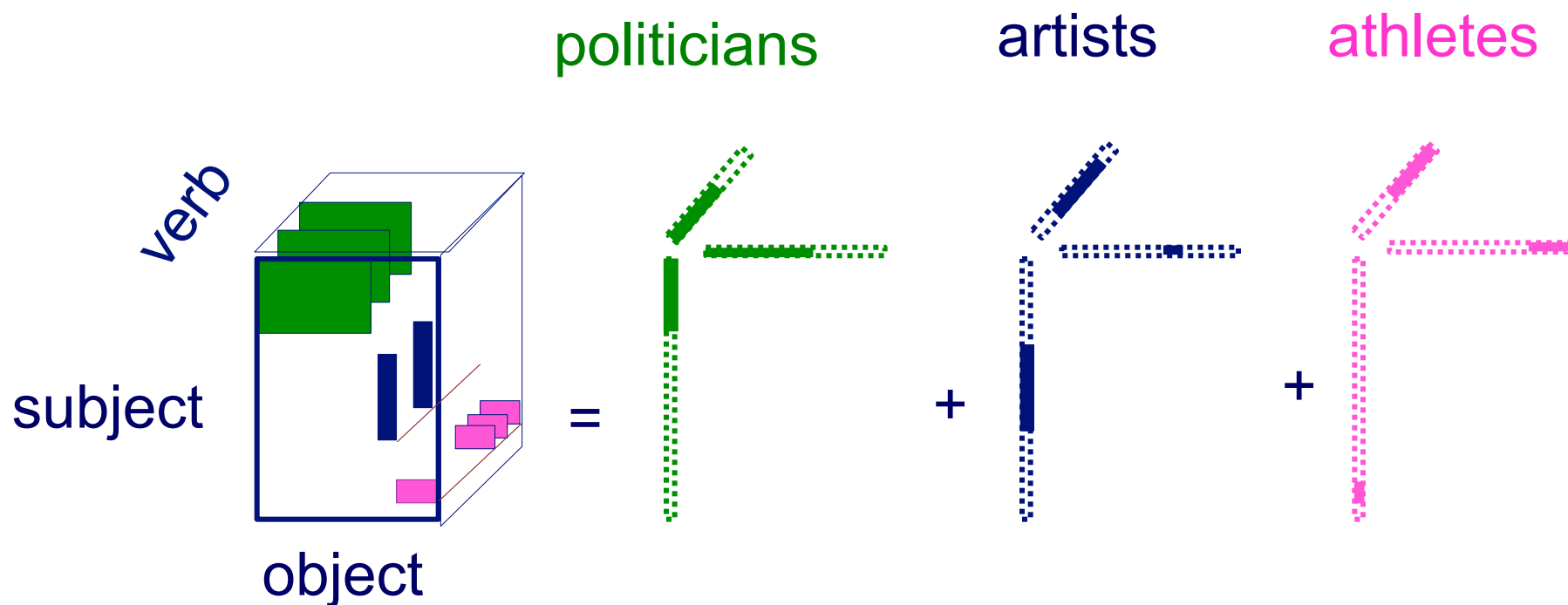
Answer to both: tensor factorization

- Recall: (SVD) matrix factorization: finds blocks



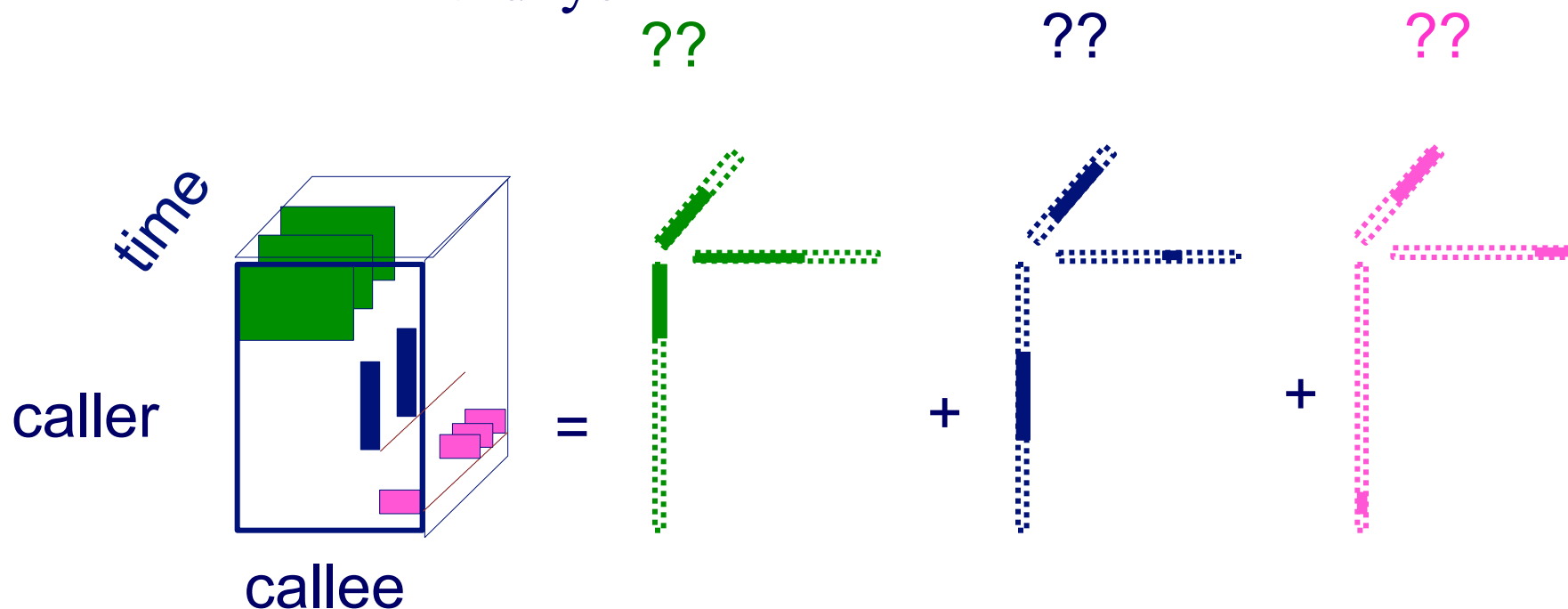
Answer to both: tensor factorization

- PARAFAC decomposition

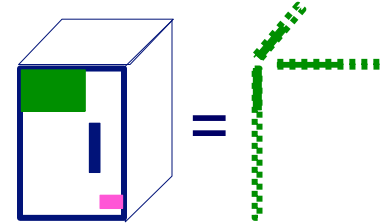


Answer: tensor factorization

- PARAFAC decomposition
- Results for who-calls-whom-when
 - 4M x 15 days

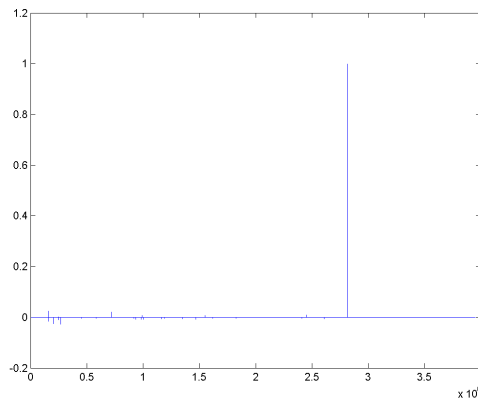


Anomaly detection in time-evolving graphs

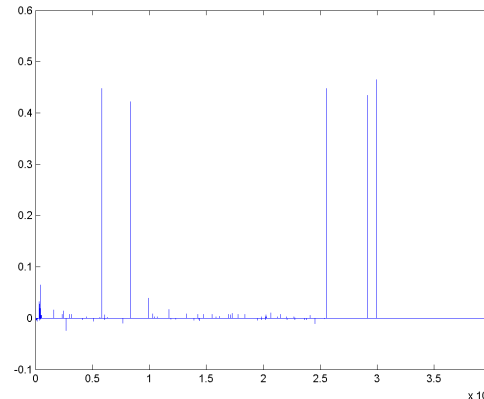


- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

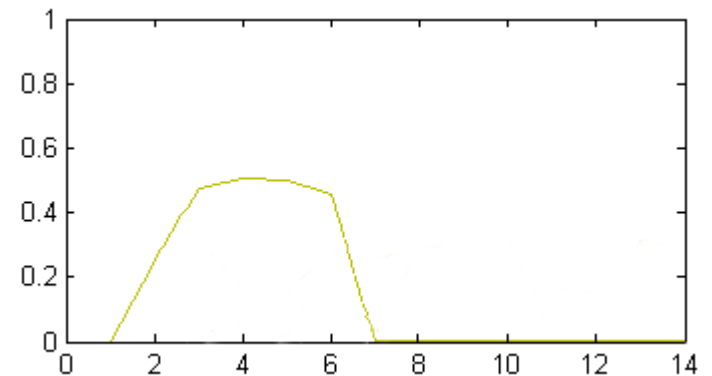
1 caller



5 receivers

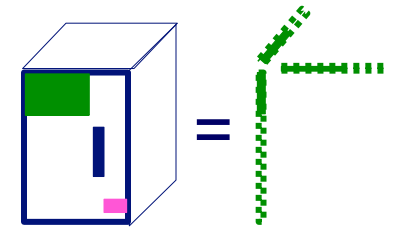


4 days of activity



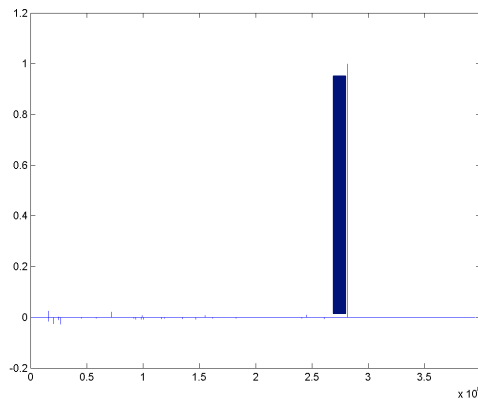
~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs

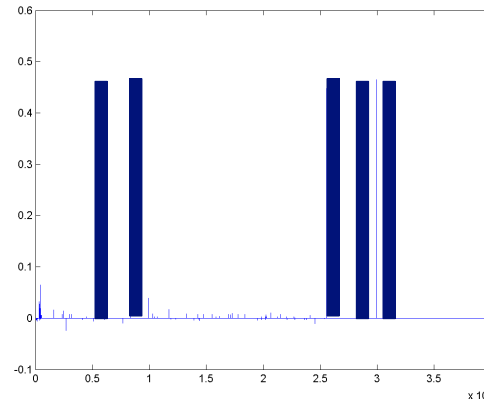


- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks

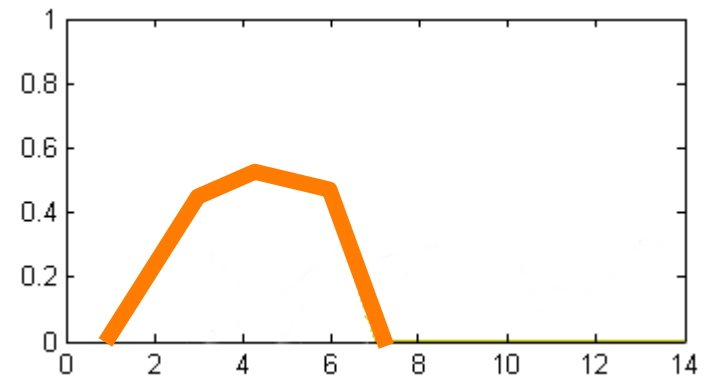
1 caller



5 receivers

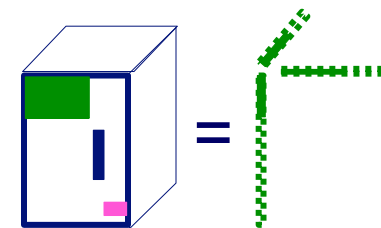


4 days of activity

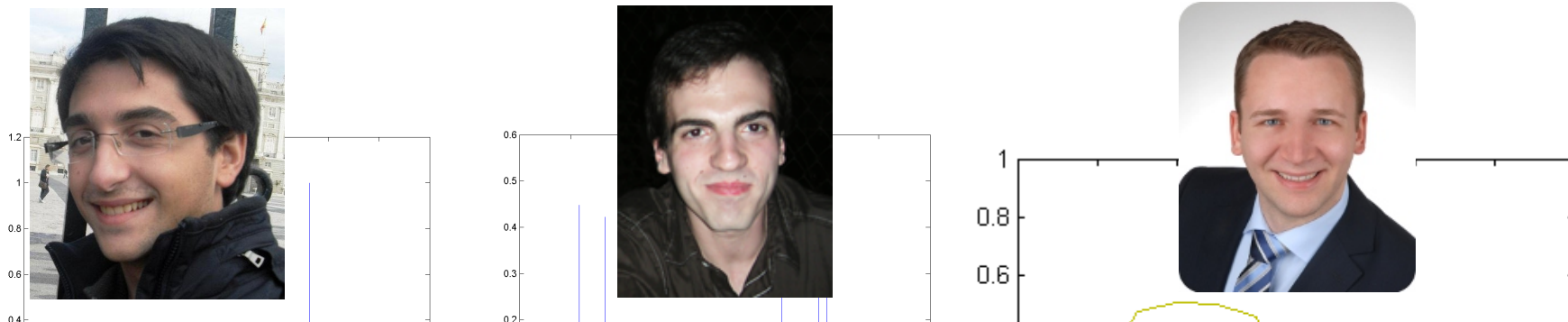


~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs



- Anomalous communities in phone call data:
 - European country, 4M clients, data over 2 weeks



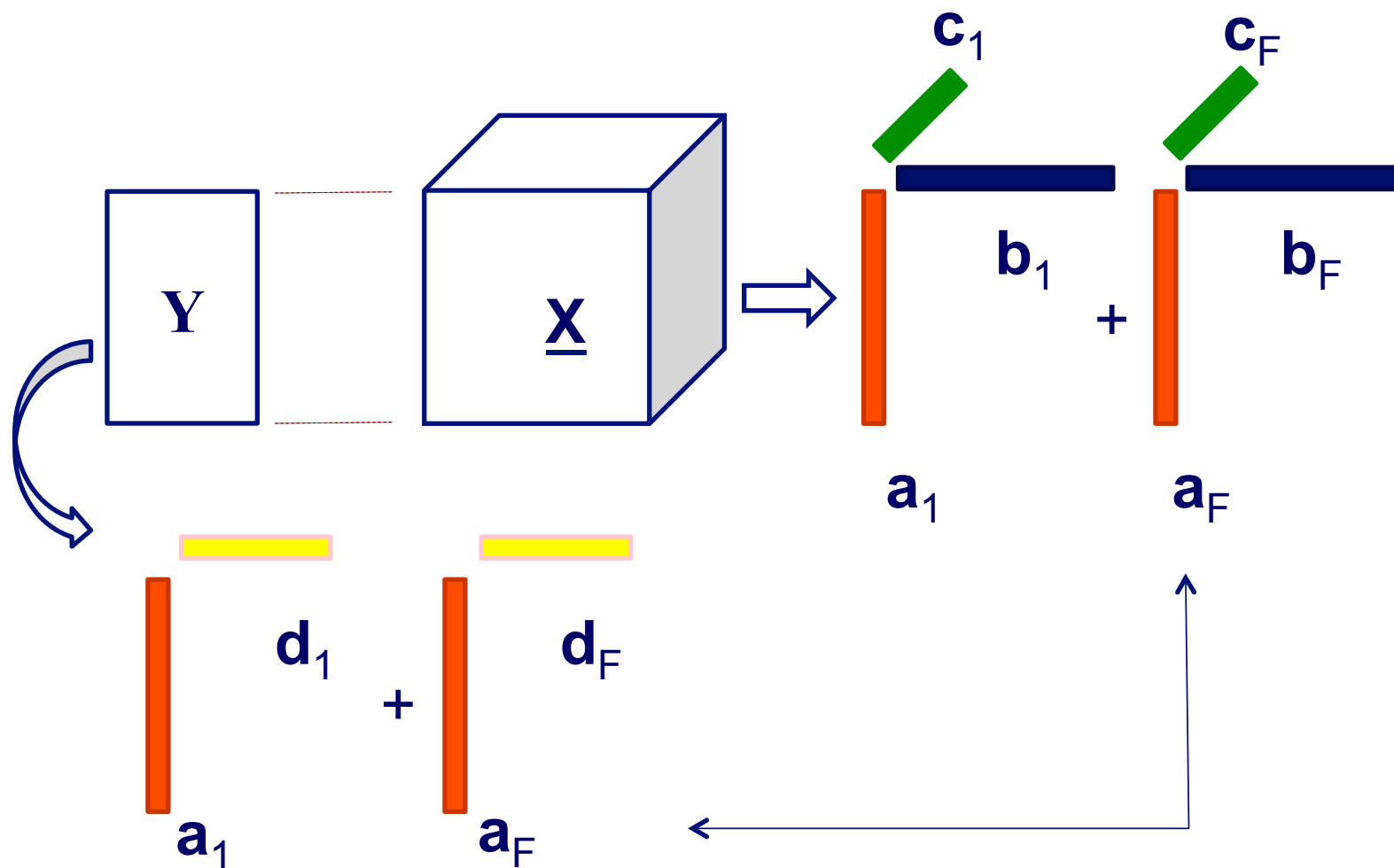
Miguel Araujo, Spiros Papadimitriou, Stephan Günnemann, Christos Faloutsos, Prithwish Basu, Ananthram Swami, Evangelos Papalexakis, Danai Koutra. *Com2: Fast Automatic Discovery of Temporal (Comet) Communities*. PAKDD 2014, Tainan, Taiwan.

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
 - P2.1: Discoveries @ phonecall network
 - ➔ – P2.2: Discoveries in neuro-semantics
 - Speed
- Conclusions

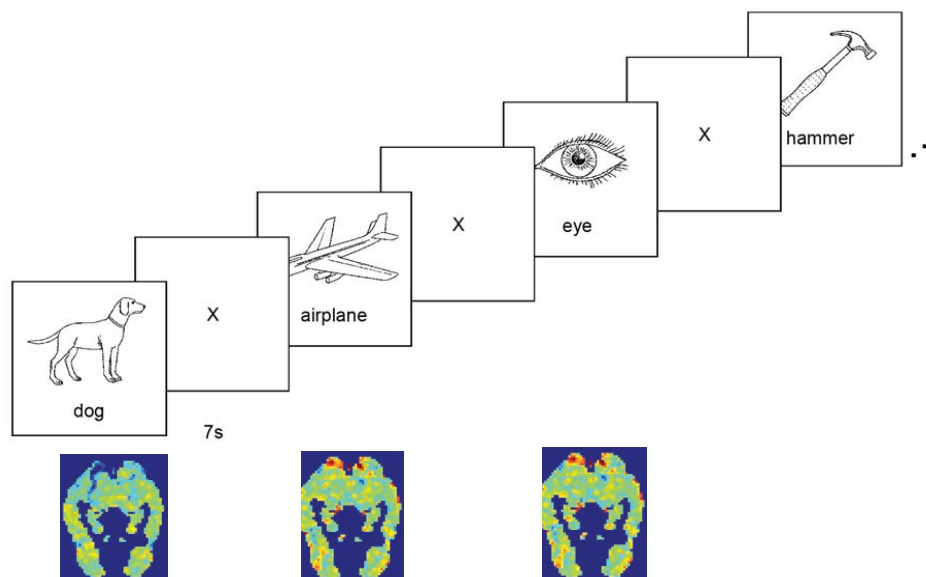


Coupled Matrix-Tensor Factorization (CMTF)



Neuro-semantic

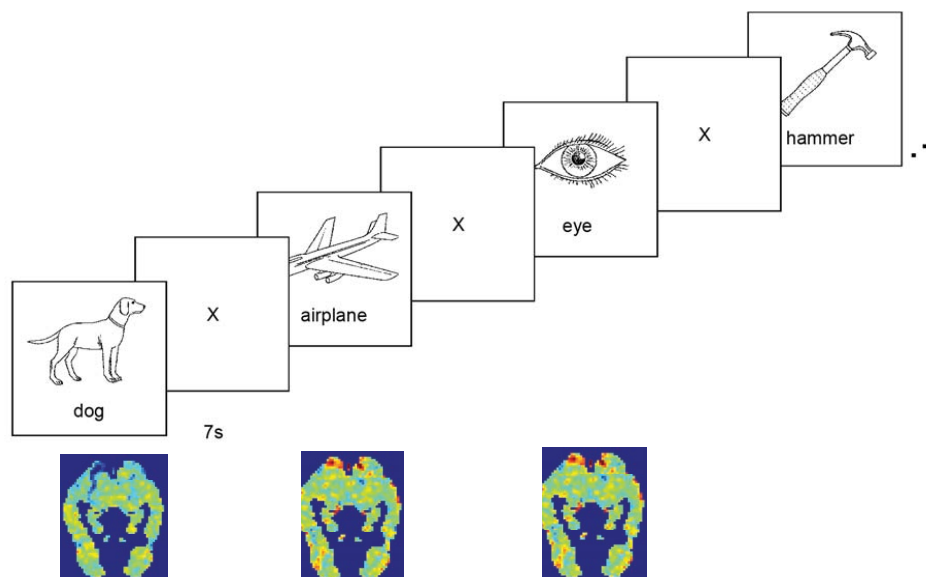
- **Brain Scan Data***
 - 9 persons
 - 60 nouns
- **Questions**
 - 218 questions
 - 'is it alive?', 'can you eat it?'



*Mitchell et al. *Predicting human brain activity associated with the meanings of nouns*. Science, 2008. Data@ www.cs.cmu.edu/afs/cs/project/theo-73/www/science2008/data.html

Neuro-semantic

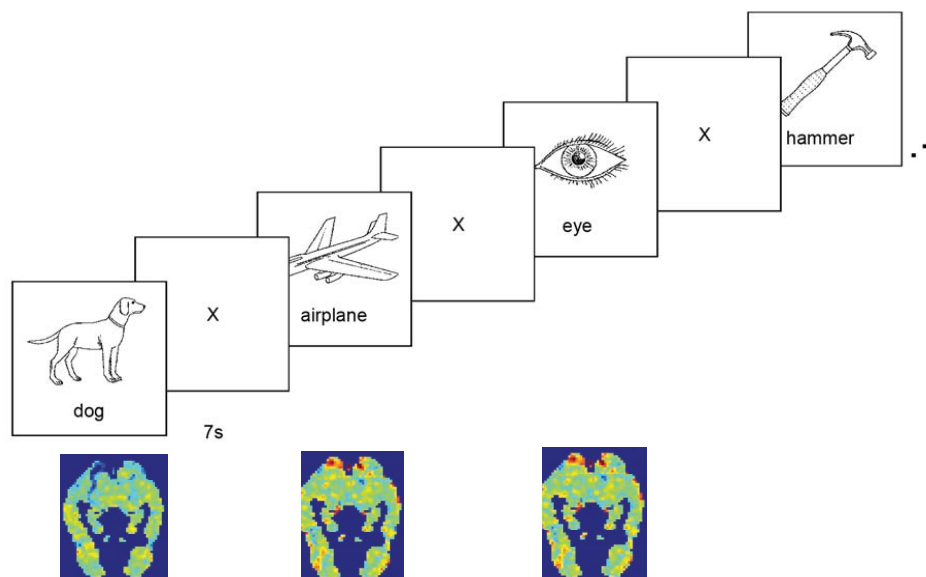
- **Brain Scan Data***
 - 9 persons
 - 60 nouns
- **Questions**
 - 218 questions
 - 'is it alive?', 'can you eat it?'



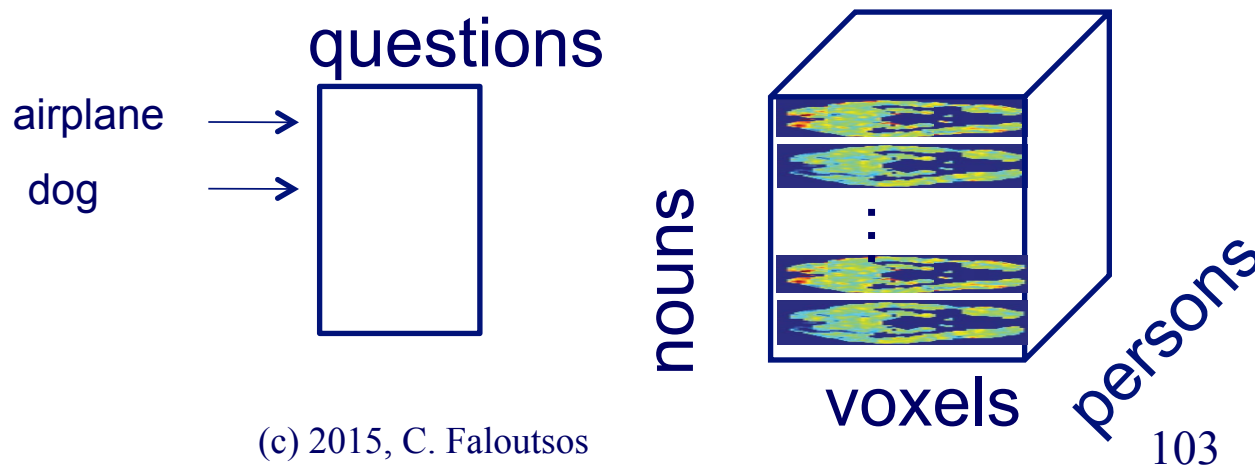
Patterns?

Neuro-semantic

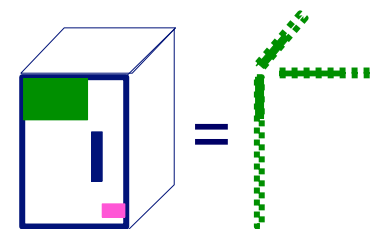
- **Brain Scan Data***
 - 9 persons
 - 60 nouns
- **Questions**
 - 218 questions
 - 'is it alive?', 'can you eat it?'



Patterns?



Neuro-semantic



Nouns

beetle
pants
bee

Questions

can it cause you pain?
do you see it daily?
is it conscious?

Nouns

bear
cow
coat

Questions

does it grow?
is it alive?
was it ever alive?

Nouns

glass
tomato
bell

Questions

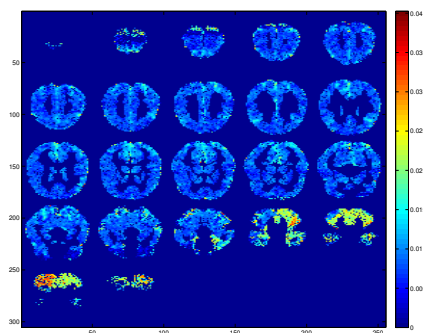
can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?

Nouns

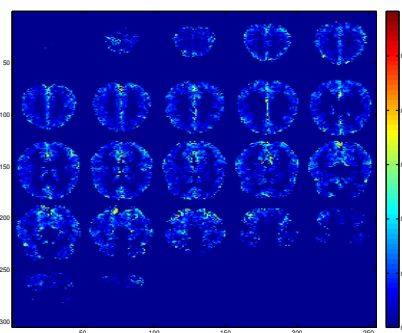
bed
house
car

Questions

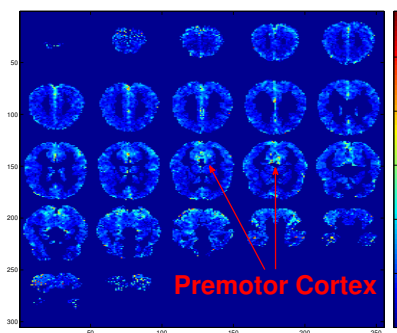
does it use electricity?
can you sit on it?
does it cast a shadow?



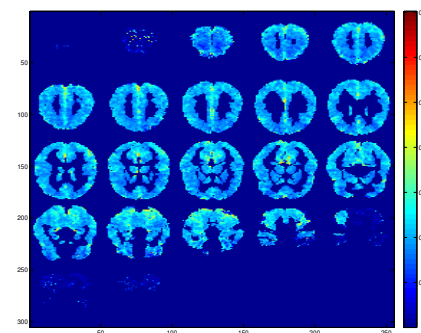
Group 1



Group 2

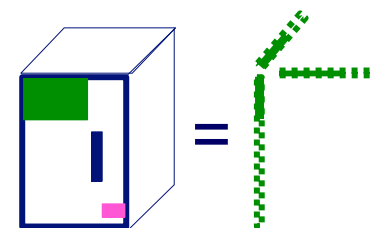


Group 3



Group 4

Neuro-semantic



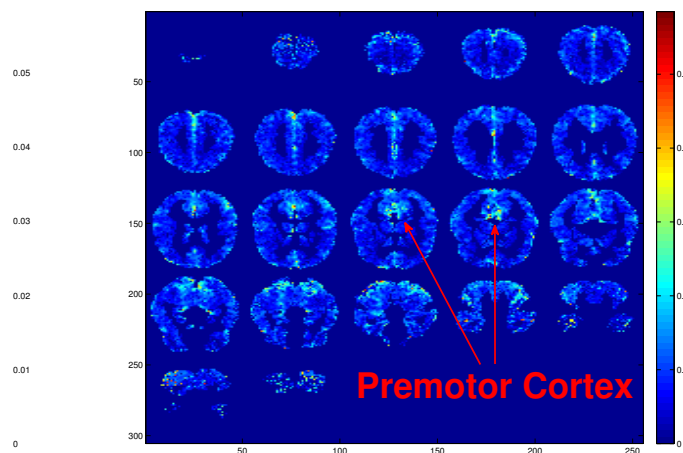
**Small items ->
Premotor cortex**

Nouns

glass
tomato
bell

Questions

can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?



Neuro-semantic

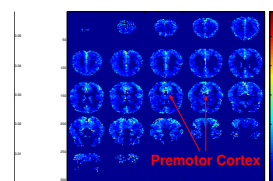
**Small items ->
Premotor cortex**

Nouns

glass
tomato
bell

Questions

can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?



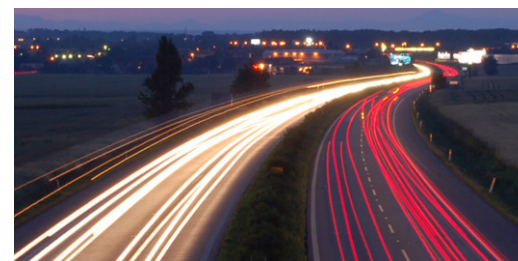
Group 3



Evangelos Papalexakis, Tom Mitchell, Nicholas Sidiropoulos,
Christos Faloutsos, Partha Pratim Talukdar, Brian Murphy,
*Turbo-SMT: Accelerating Coupled Sparse Matrix-Tensor
Factorizations by 200x*, SDM 2014

Roadmap

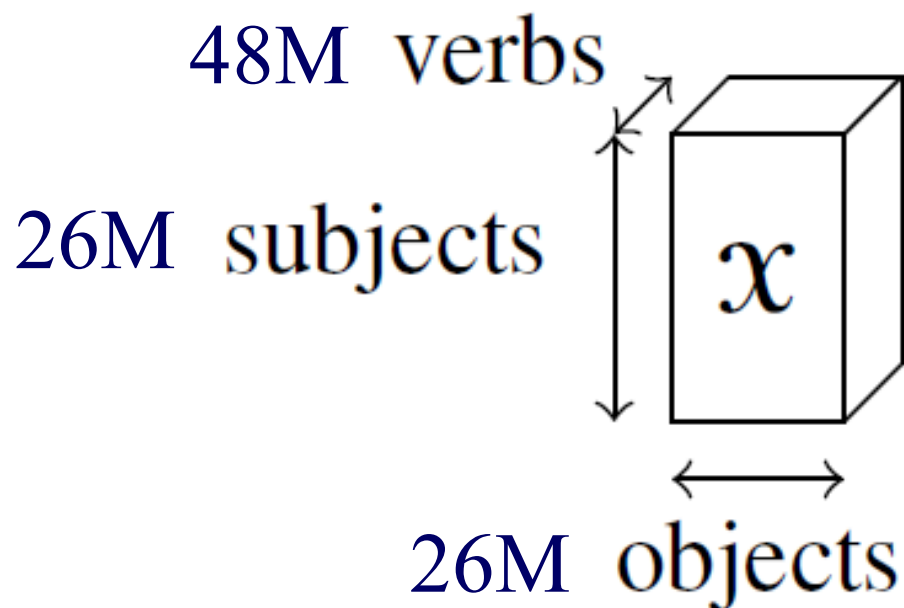
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
 - P2.1: Discoveries @ phonecall network
 - P2.2: Discoveries in neuro-semantics
- ➔ – Scalability & Speed
- Conclusions



Q: spilling to the disk?

Reminder: tensor (eg., Subject-verb-object)

144M non-zeros

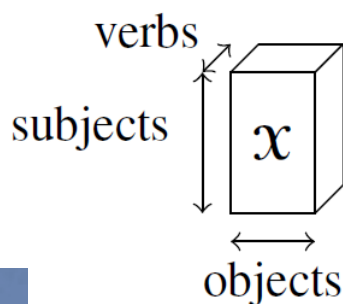


NELL (Never Ending
Language Learner)
@CMU

A: GigaTensor

Reminder: tensor (eg., Subject-verb-object)

26M x 48M x 26M, 144M non-zeros



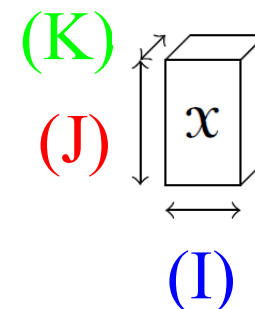
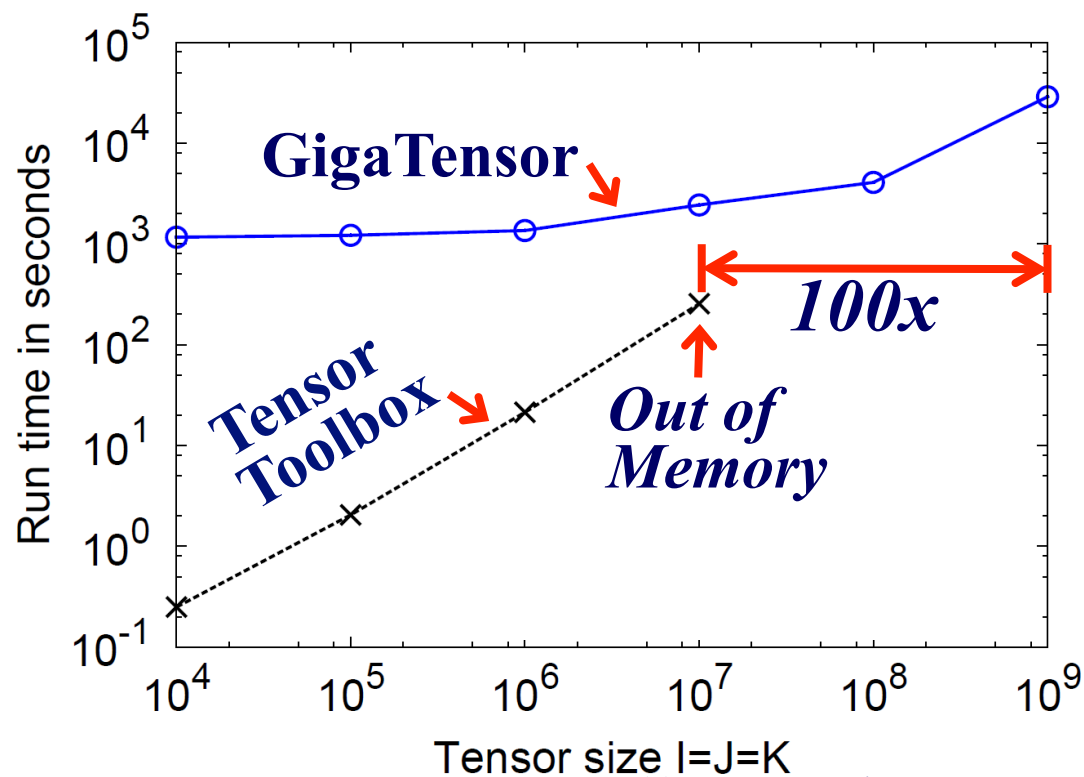
NELL (Never Ending
Language Learner)
@CMU



U Kang, Evangelos E. Papalexakis, Abhay Harpale, Christos Faloutsos, *GigaTensor: Scaling Tensor Analysis Up By 100 Times - Algorithms and Discoveries*, KDD'12

A: GigaTensor

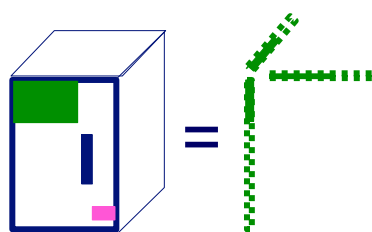
- GigaTensor solves *100x* larger problem



Number of
nonzero
= $I / 50$

Part 2: Conclusions

- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- GigaTensor -> fast & scalable



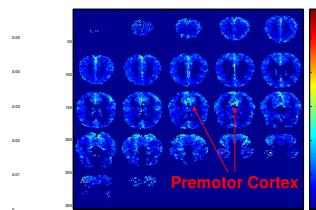
UIC

Nouns

glass
tomato
bell

Questions

can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?



Group 3

Roadmap

- Introduction – Motivation
 - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Part#3: Cascades and immunization
- Future directions
- ➔ • Acknowledgements and Conclusions



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

Cast



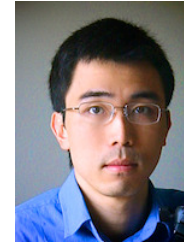
Akoglu,
Leman



Araujo,
Miguel



Beutel,
Alex



Chau,
Polo



Kang, U



Koutra,
Danai



Lee,
Jay Yoon



Prakash,
Aditya



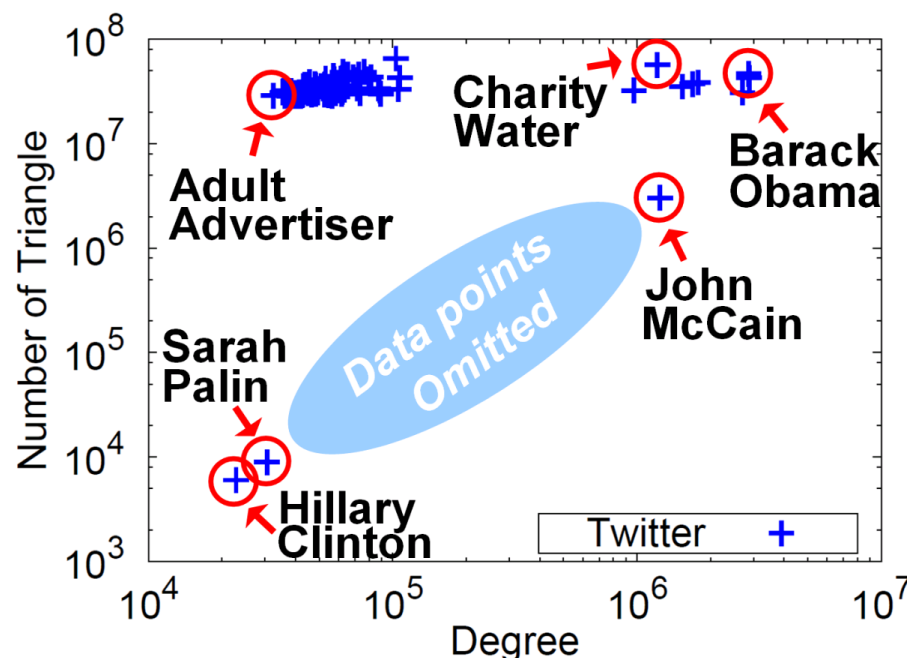
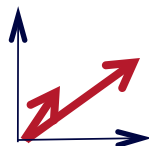
Papalexakis,
Vagelis



Shah,
Neil

CONCLUSION#1 – Big data

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



CONCLUSION#2 – tensors

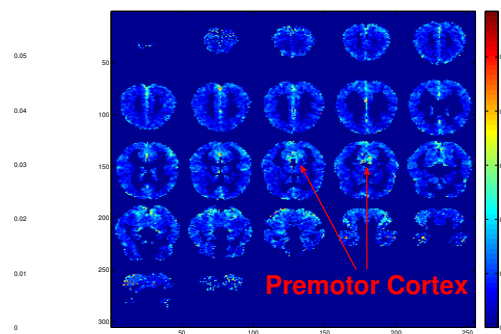
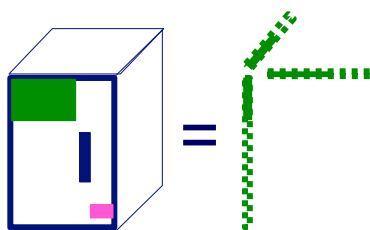
- powerful tool

Nouns

glass
tomato
bell

Questions

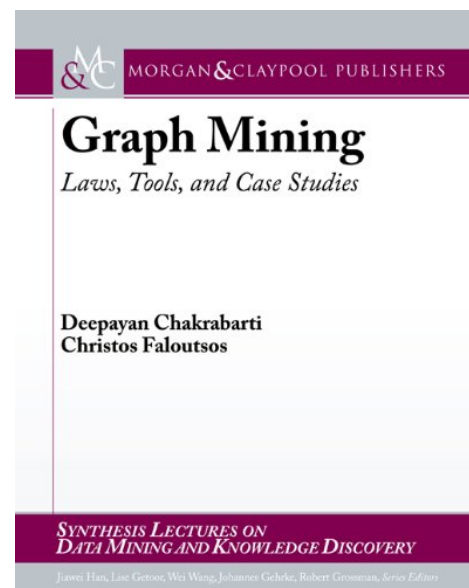
can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?



Group 3

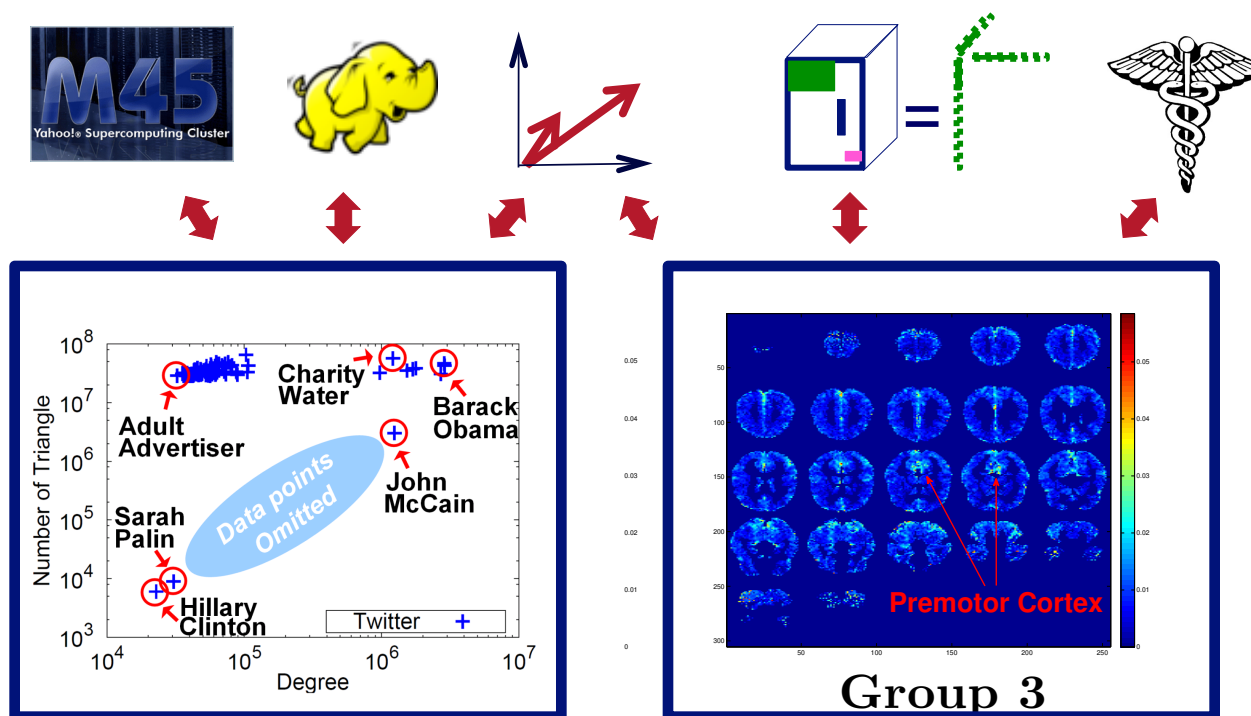
References

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



TAKE HOME MESSAGE:

Cross-disciplinary



Thank you!

Cross-disciplinary

