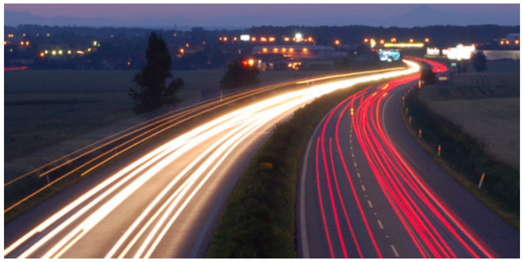
Carnegie Mellon

Mining Large Graphs

Christos Faloutsos
CMU

Carnegie Mellon

Roadmap

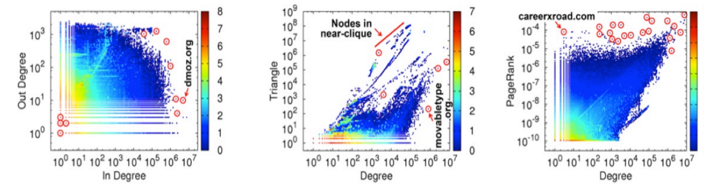


- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs
- Conclusions

Graphs - why should we care?

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

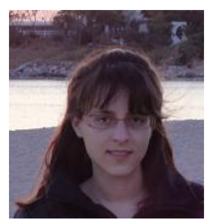
Graphs - why should we care?



YahooWeb:
(a) In-degree vs. Out-degree (b) Degree vs. Triangles (c) Degree vs. PageRank

Twitter:
(d) In-degree vs. Out-degree (e) Degree vs. Triangles (f) Degree vs. PageRank

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

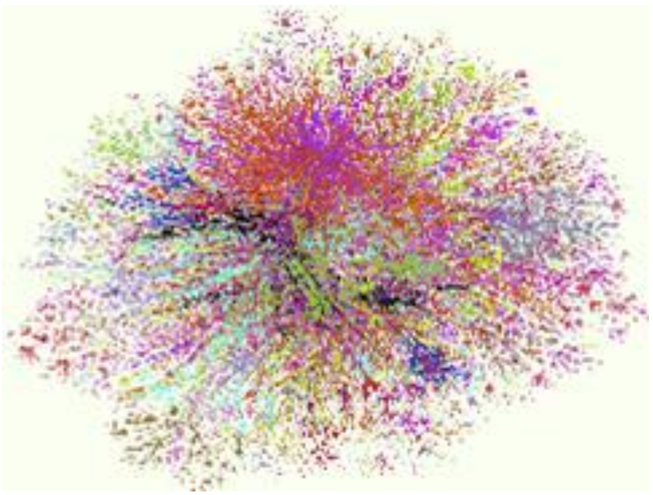


U Kang, Jay-Yoon Lee, Danai Koutra, and Christos Faloutsos. *Net-Ray: Visualizing and Mining Billion-Scale Graphs* PAKDD 2014, Tainan, Taiwan.

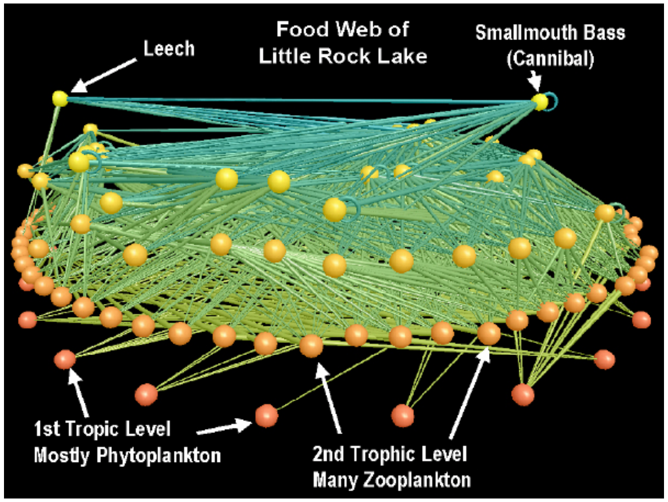
Graphs - why should we care?

>$10B; ~1B users

Graphs - why should we care?



Internet Map
[lumeta.com]



Food Web
[Martinez '91]

Graphs - why should we care?

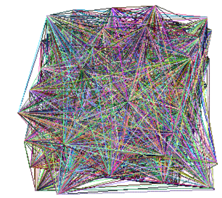
- web-log (‘blog’) news propagation
- computer network security: email/IP traffic and anomaly detection
- Recommendation systems
- ....

- Many-to-many db relationship -> graph

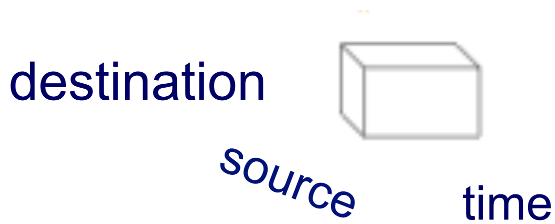
Carnegie Mellon

Motivating problems

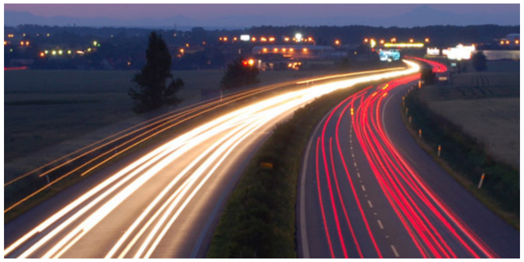
- P1: patterns? Fraud detection?



- P2: patterns in time-evolving graphs / tensors

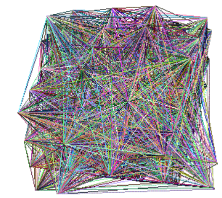


Roadmap



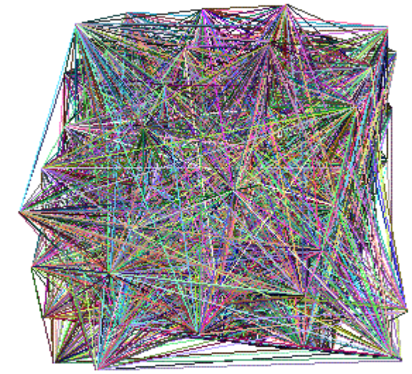
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs
- Conclusions

Carnegie Mellon



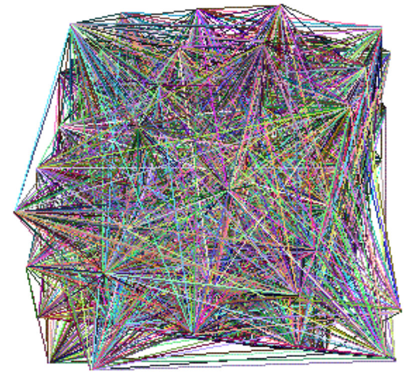
Part 1: Patterns, & fraud detection

Laws and patterns

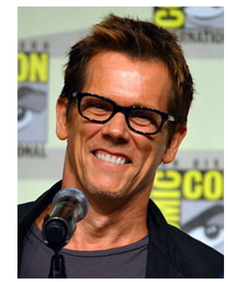


- Q1: Are real graphs random?

Laws and patterns



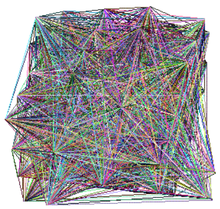
- Q1: Are real graphs random?
- A1: NO!!
  - Diameter (‘6 degrees’; ‘Kevin Bacon’)
  - in- and out- degree distributions
  - other (surprising) patterns
- So, let’s look at the data

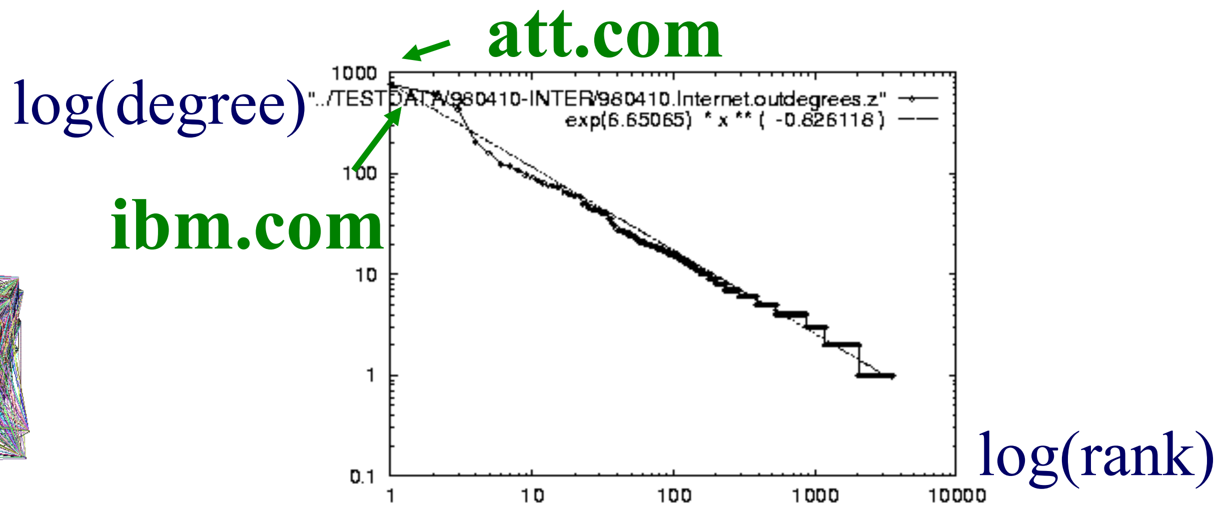


Solution# S.1

- Power law in the degree distribution [FFF, SIGCOMM99]

internet domains

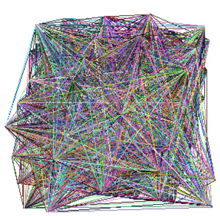


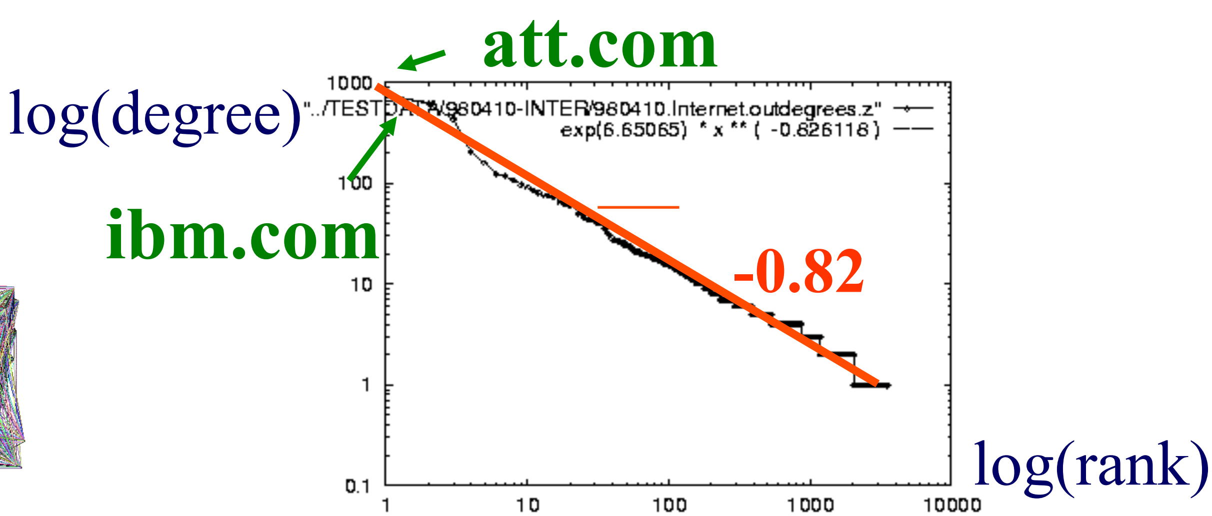


Solution# S.1

- Power law in the degree distribution [FFF, SIGCOMM99]

internet domains

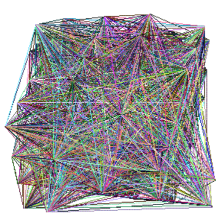


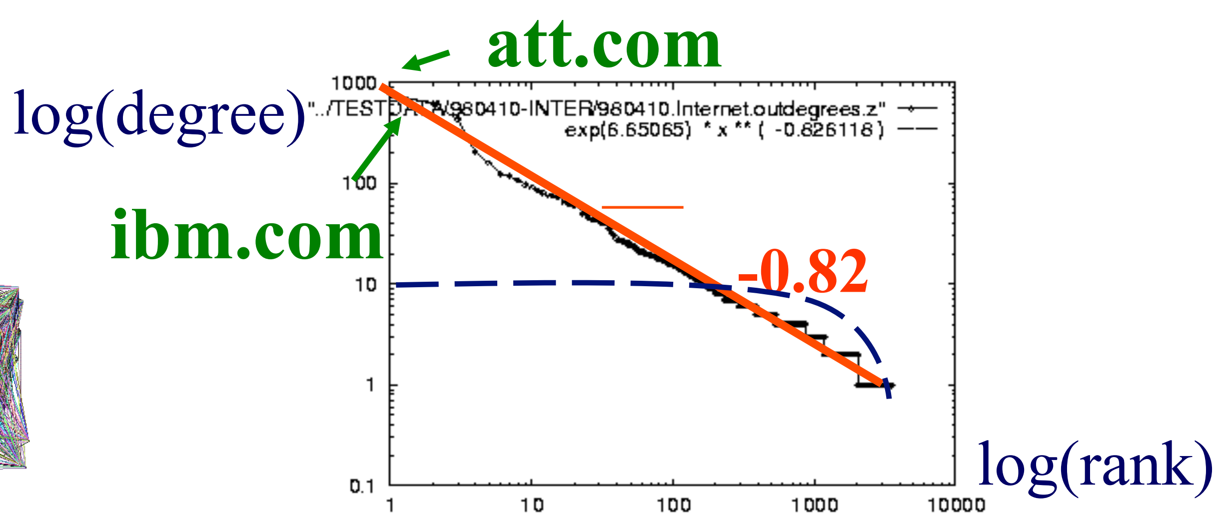


Solution# S.1

- Q: So what?

internet domains



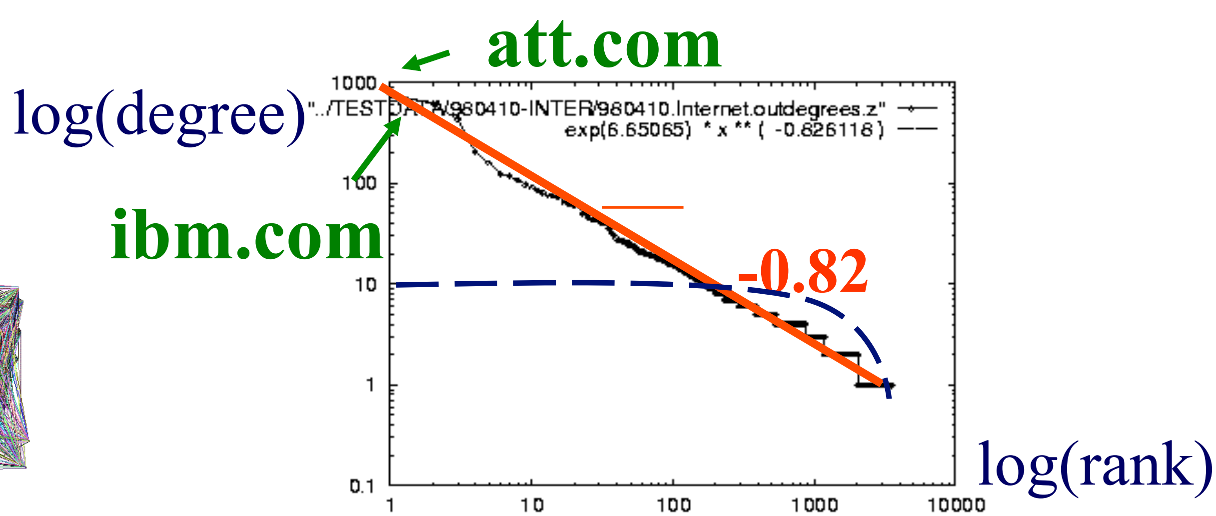


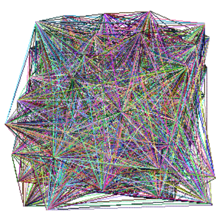
Solution# S.1

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

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

internet domains



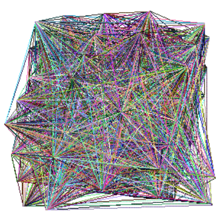


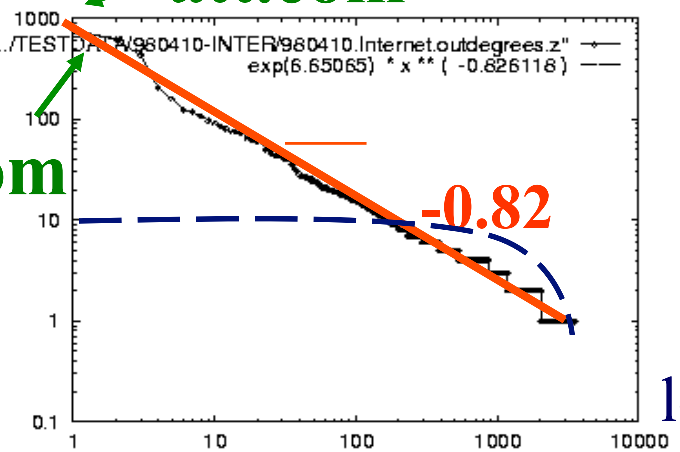
Solution# S.1

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

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

internet domains





log(degree)

att.com

ibm.com

-0.82

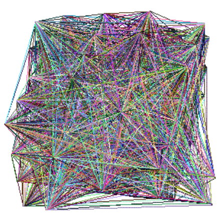
log(rank)

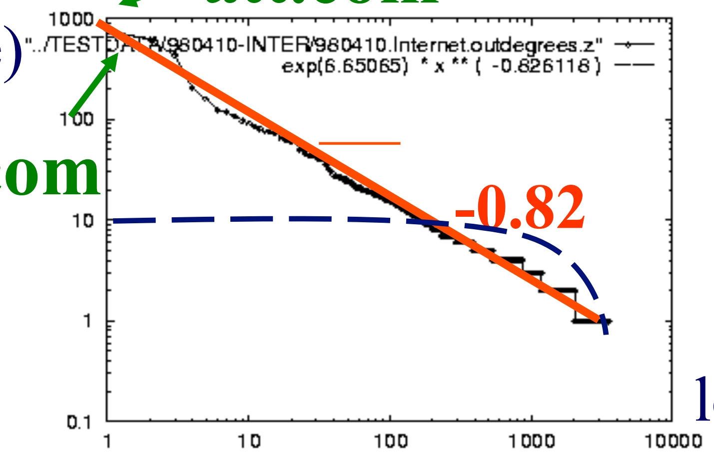
Solution# S.1

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

- Q: So what?
- A1: # of two-step-away pairs: 100^2 Trillion

internet domains





Gaussian trap

Solution# S.1

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

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

~0.8PB ->
a data center(!)

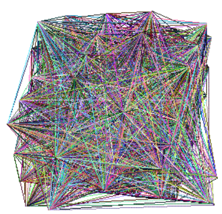
internet domains

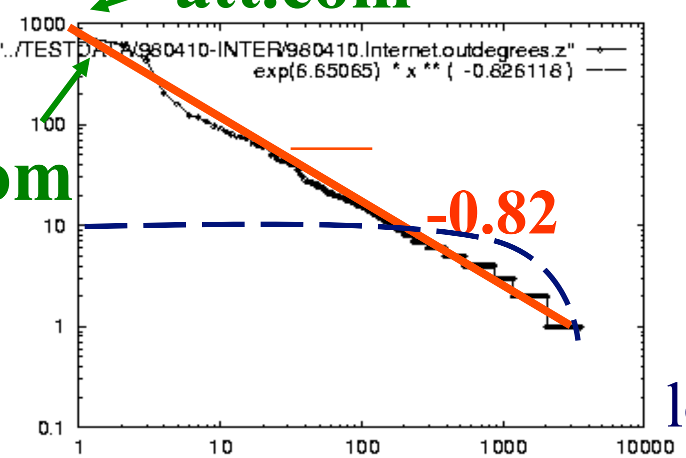
att.com

log(degree)

ibm.com

-0.82





DCO @ CMU

Gaussian trap

Solution# S.1

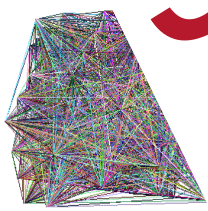
- Q: So what?
- A1: # of two-step-aw... ?) ~ 10M^2

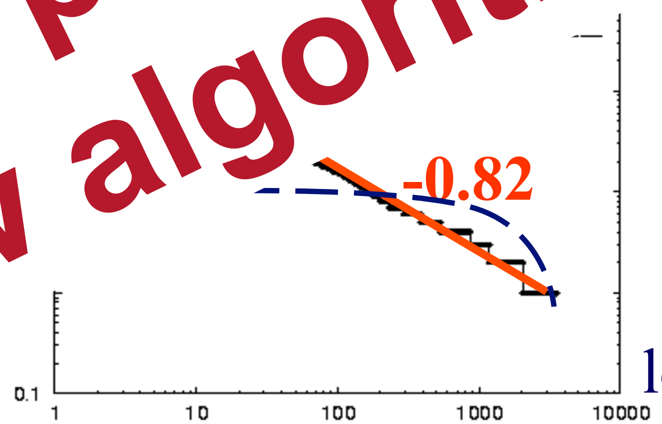
inte...

⇩

~0.8PB ->
a data center(!)

Such patterns ->
New algorithms



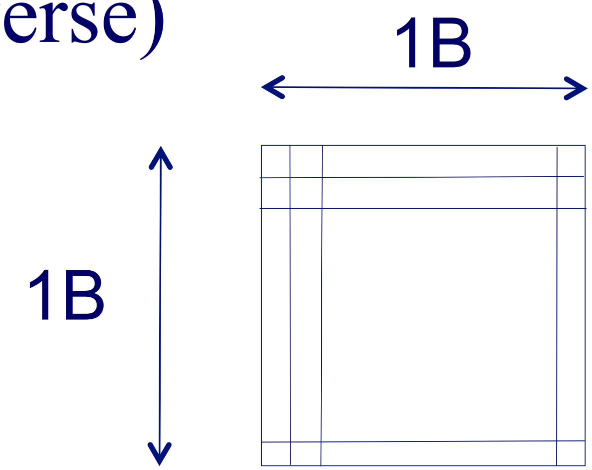


Carnegie Mellon

Observation – big-data:

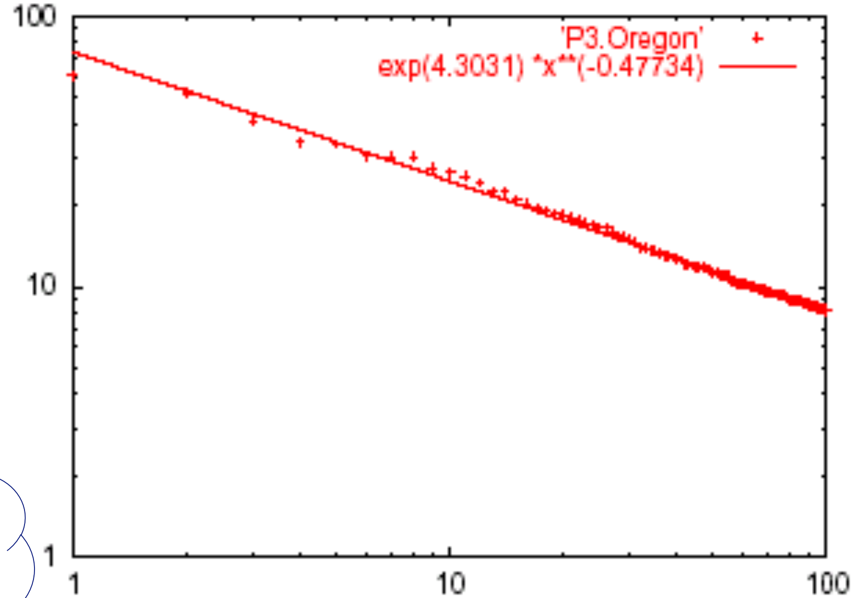
- O(N^2) algorithms are ~intractable - N=1B

- N^2 seconds = 31B years (>2x age of universe)



Solution# S.2: Eigen Exponent *E*

Eigenvalue



Rank of decreasing eigenvalue

Exponent = slope

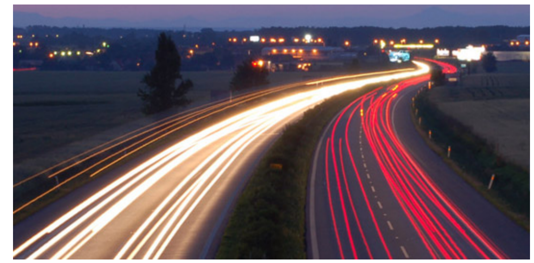
E = -0.48

May 2001

$A\,x = \lambda\,x$

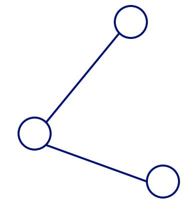
- A2: power law in the eigenvalues of the adjacency matrix ('eig()')

Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns: Degree; Triangles
  - Anomaly/fraud detection
  - Graph understanding
- Part#2: time-evolving graphs; tensors
- Part#3: Cascades and immunization
- Conclusions

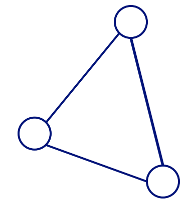
Solution# S.3: Triangle 'Laws'



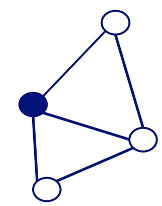
- Real social networks have a lot of triangles

Carnegie Mellon

Solution# S.3: Triangle 'Laws'



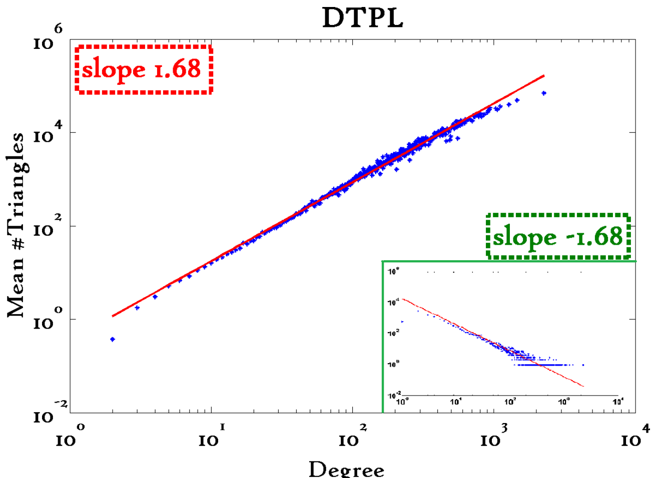
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?



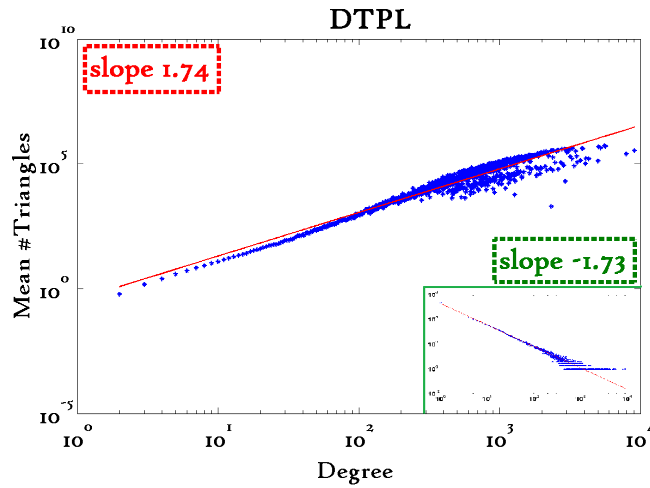
Carnegie Mellon

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

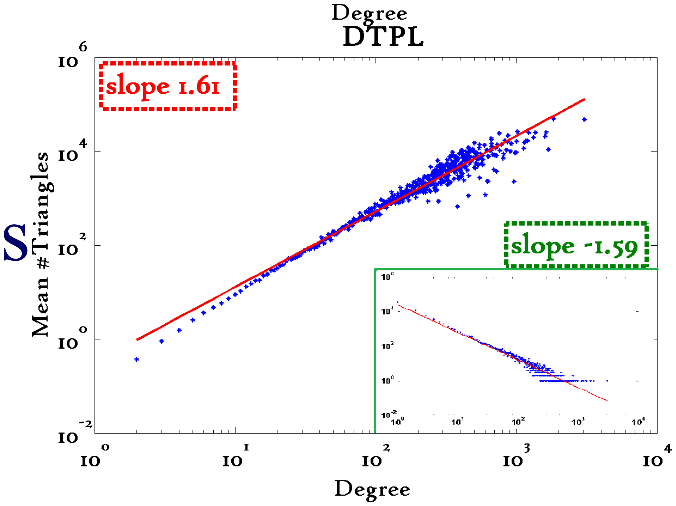
Reuters

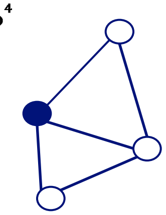


SN



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?

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

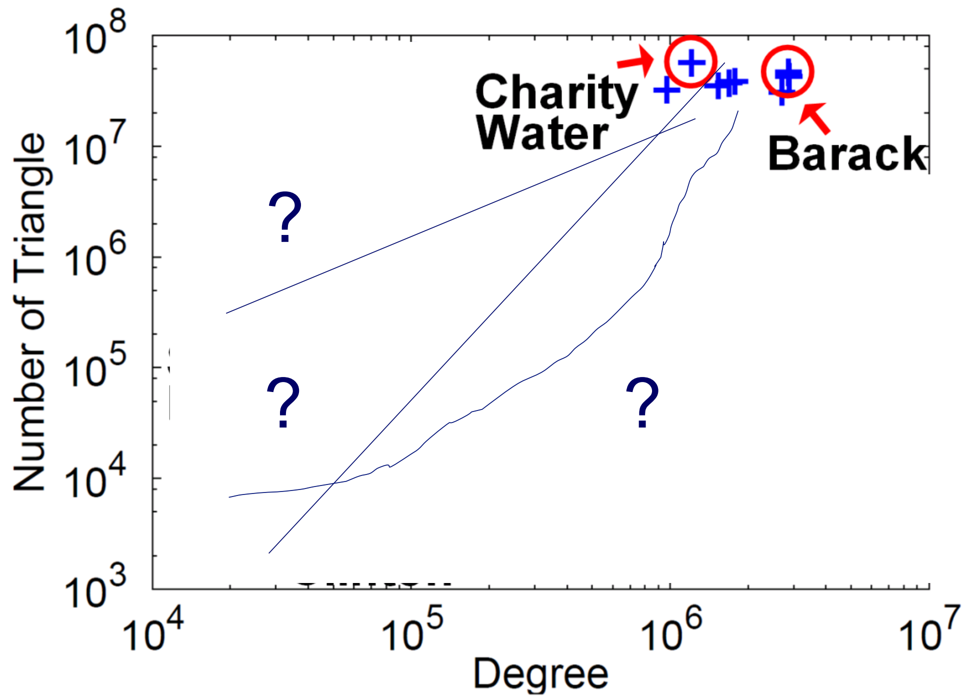
A: Yes!

#triangles = 1/6 Sum (λ_i^3)

(and, because of skewness (S2) ,

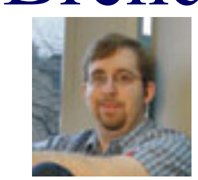
we only need the top few eigenvalues! - O(E)

Triangle counting for large graphs?

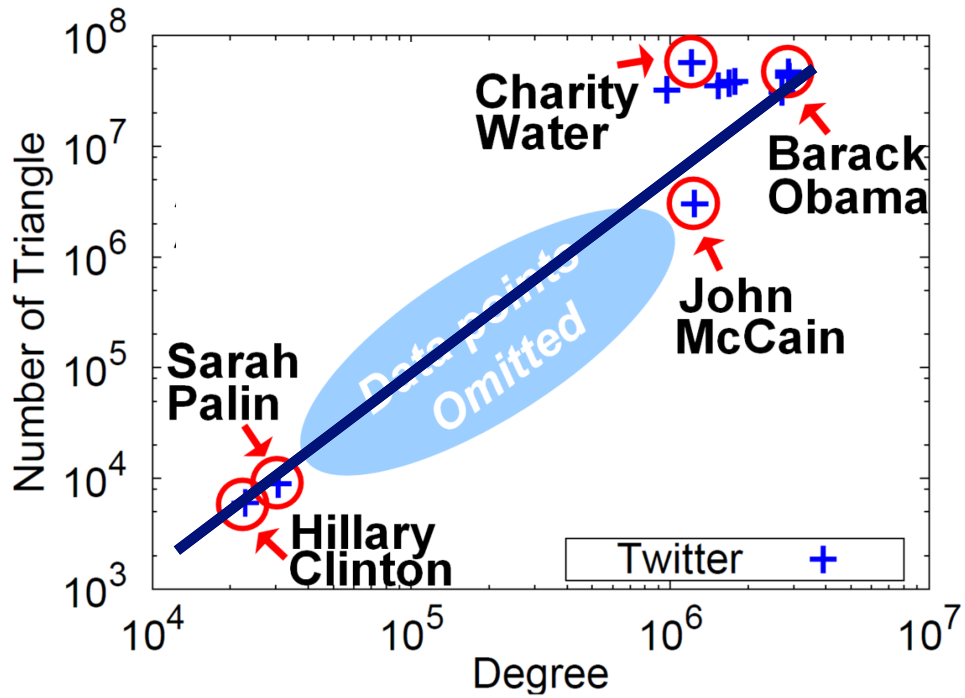


Anomalous nodes in Twitter(~ 3 billion edges)

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



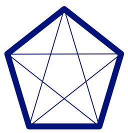
Triangle counting for large graphs?

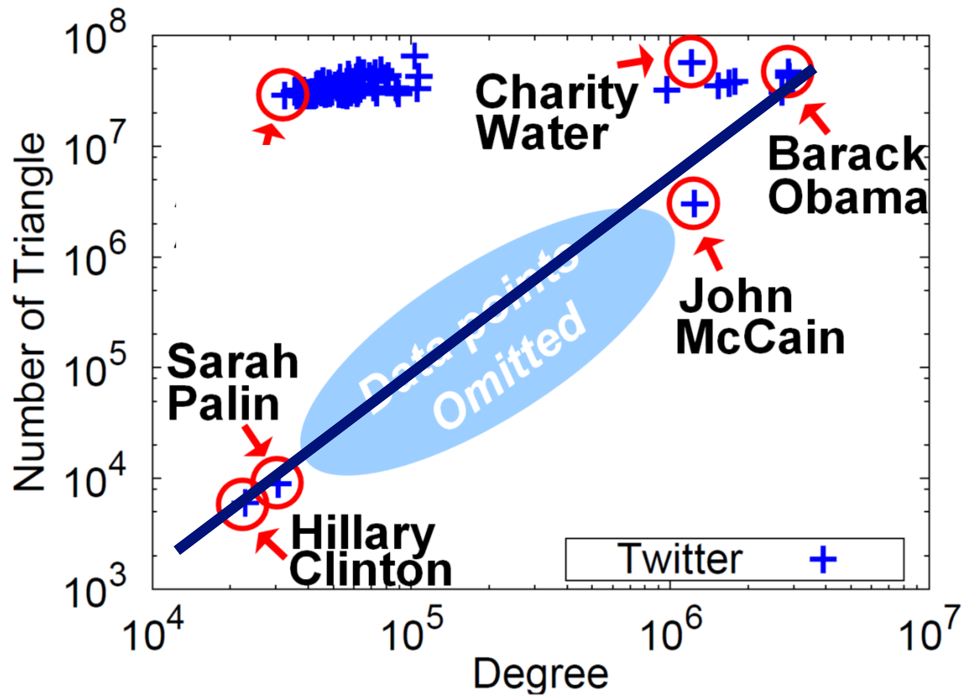


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?

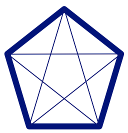


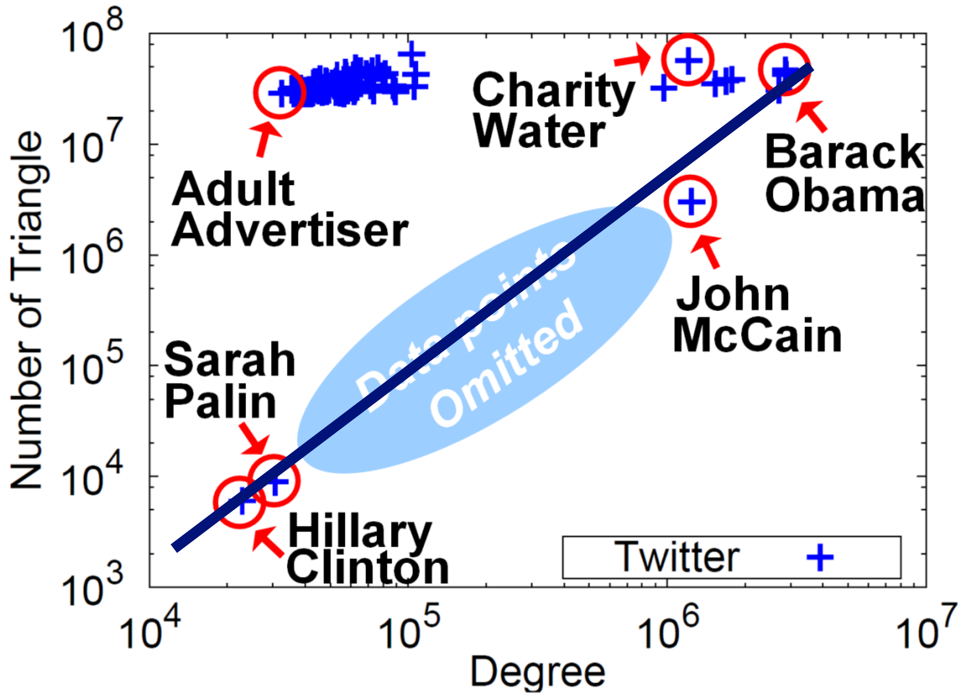


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?





Anomalous nodes in Twitter(~ 3 billion edges)

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

MORE Graph Patterns

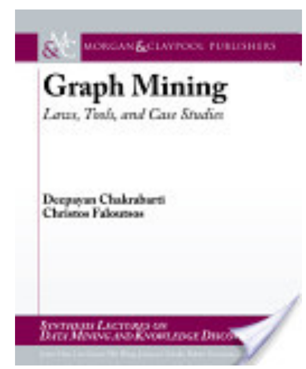
	Unweighted	Weighted
Static	✔ L01. Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04] ✔ L02. Triangle Power Law (TPL) [Tsourakakis `08] ✔ L03. Eigenvalue Power Law (EPL) [Siganos et al. `03] L04. Community structure [Flake et al. `02, Girvan 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 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 Bestavros `99, McGlohon et al. `08]	L11. Weight Power Law (WPL) [McGlohon et al. `08]

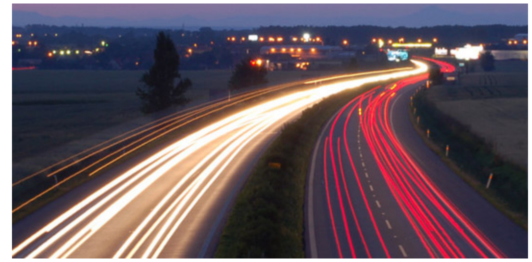

- Mary McGlohon, Leman Akoglu, Christos Faloutsos. *Statistical Properties of Social Networks.* in "Social Network Data Analytics" (Ed.: Charu Aggarwal)

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

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns
  - Anomaly / fraud detection
  - Graph understanding
- Part#2: time-evolving graphs; tensors
- Part#3: Cascades and immunization
- Conclusions

Carnegie Mellon

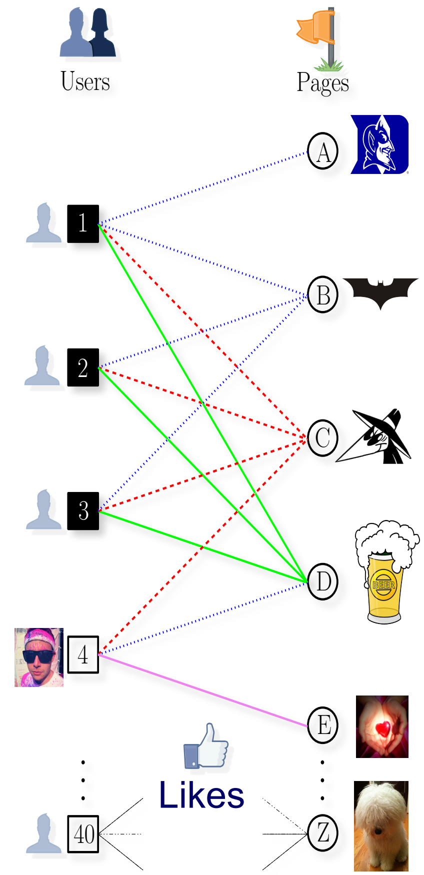
Fraud

- 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

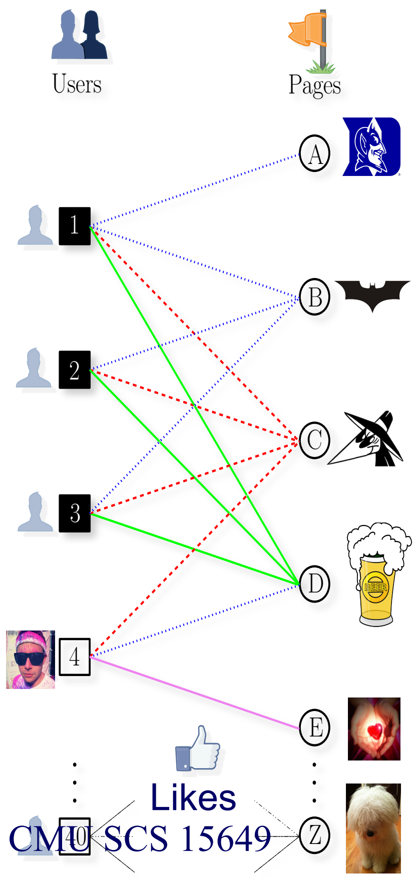


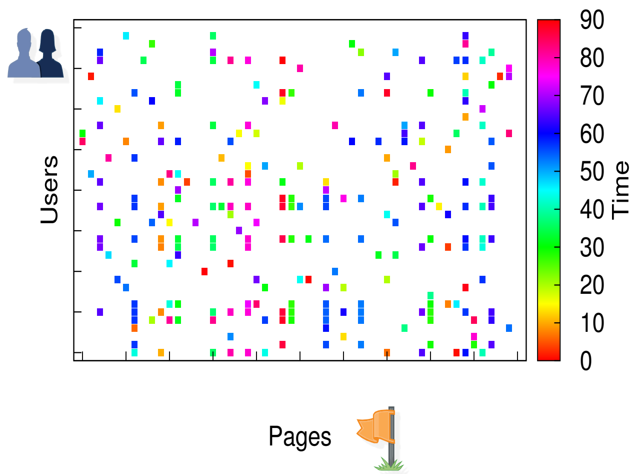
CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks, Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, Christos Faloutsos *WWW, 2013.*

Graph Patterns and Lockstep Behavior

Our intuition

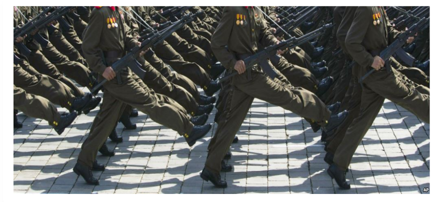
- Lockstep behavior: Same Likes, same time



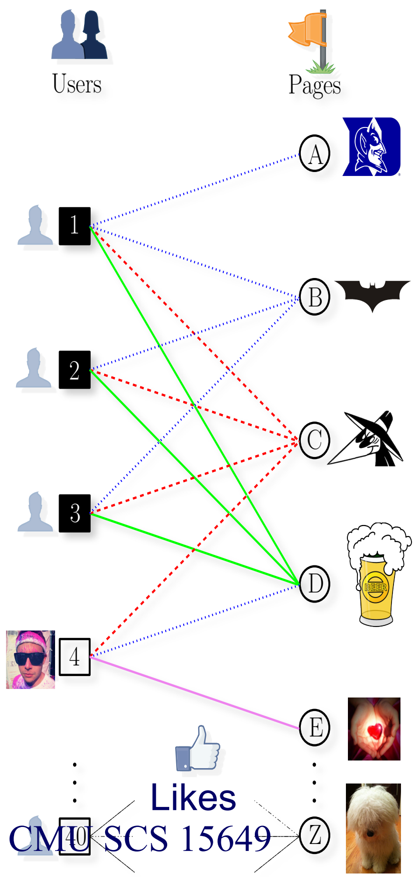


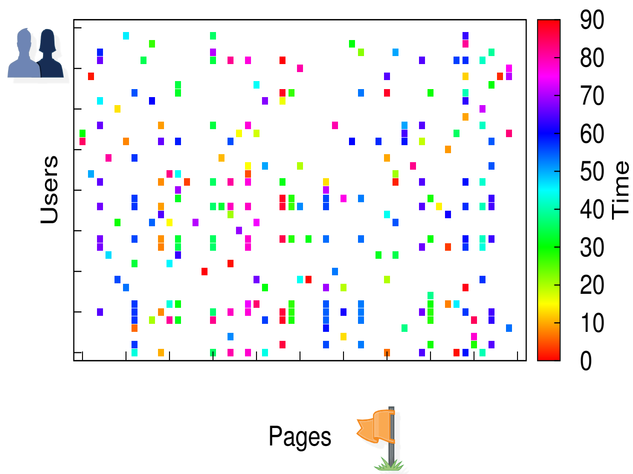
Graph Patterns and Lockstep Behavior

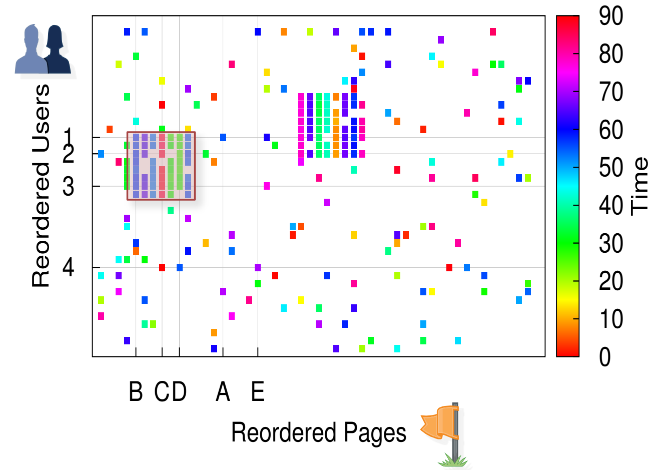
Our intuition



- Lockstep behavior: Same Likes, same time



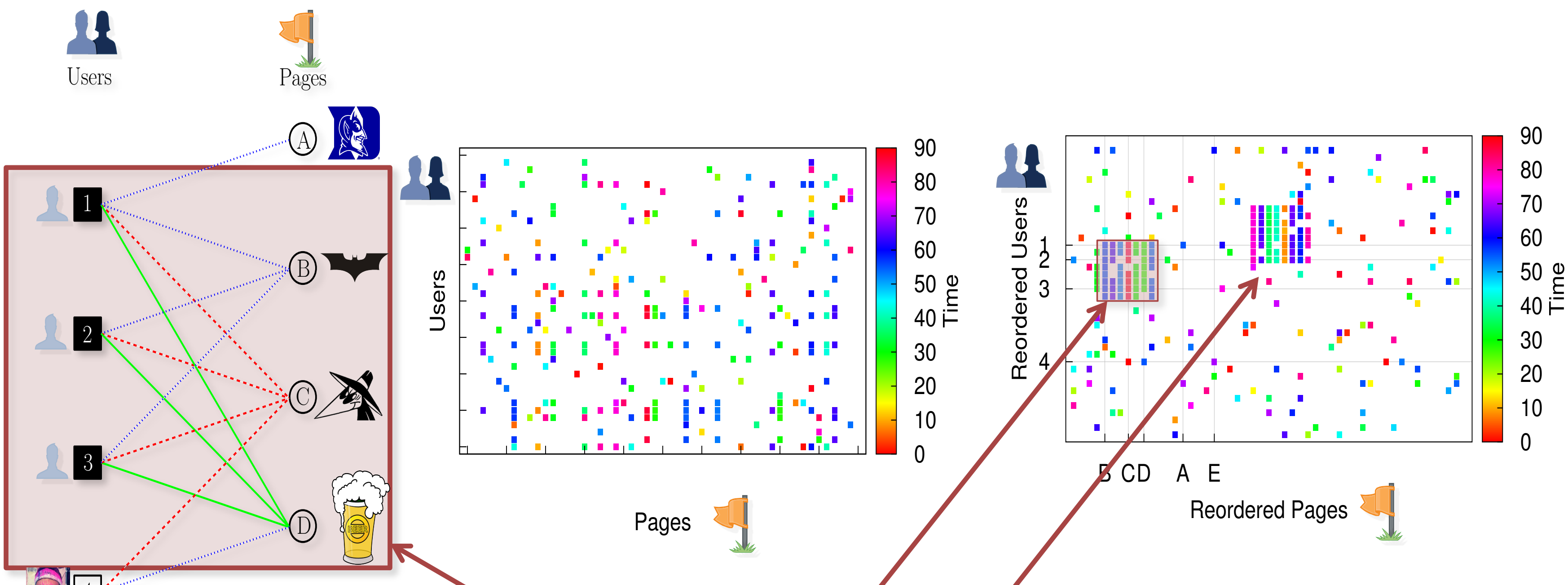




Graph Patterns and Lockstep Behavior

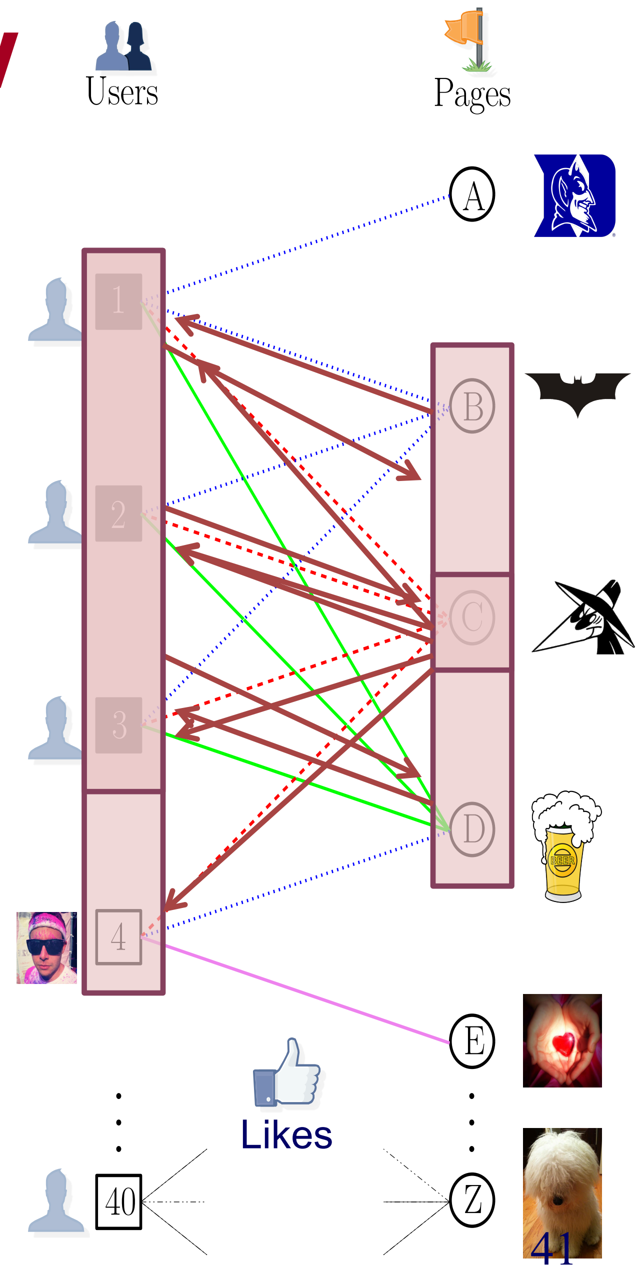
Our intuition

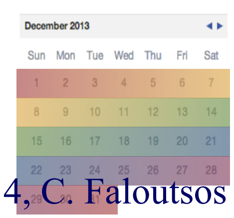
- Lockstep behavior: Same Likes, same time



MapReduce Overview

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





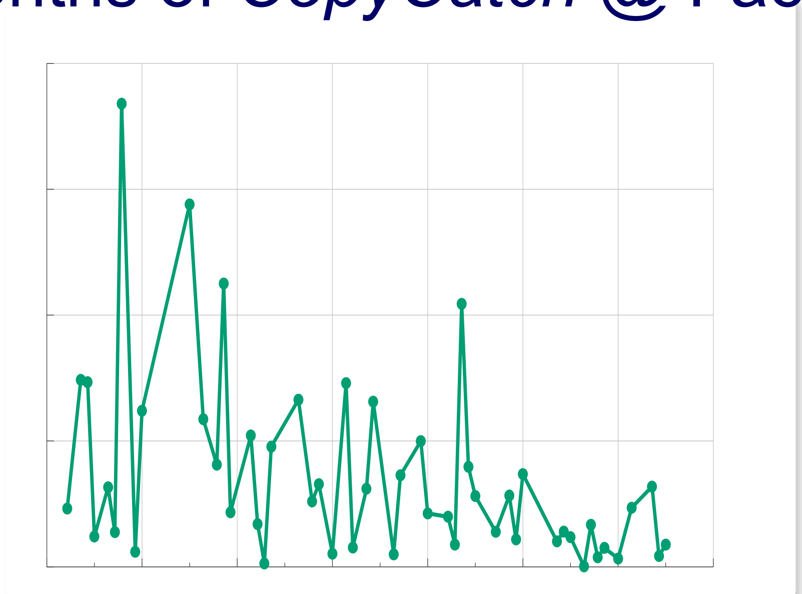
Carnegie Mellon

Deployment at Facebook

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

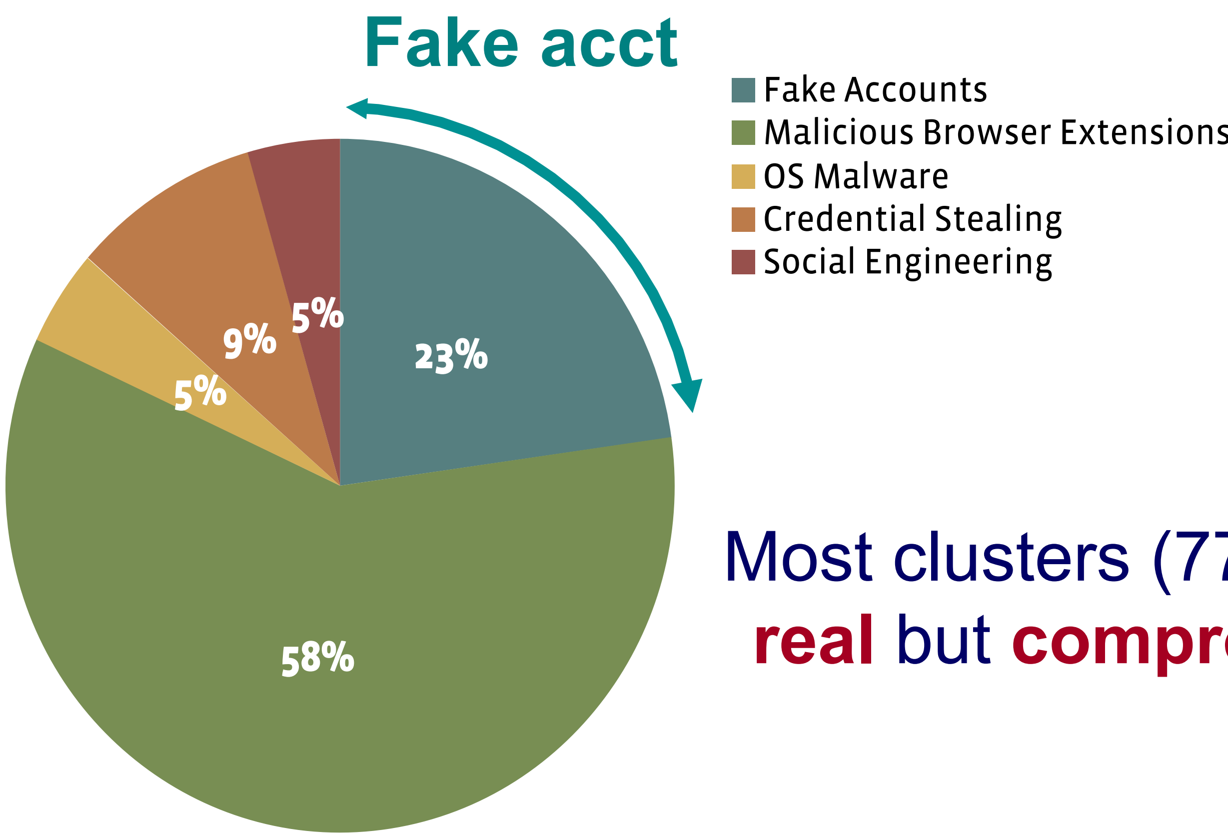
3 months of *CopyCatch* @ Facebook

#users caught



time

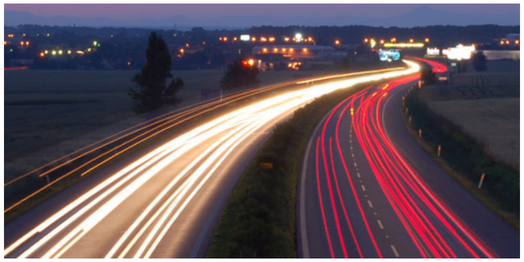
Deployment at Facebook



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

Manually labeled 22 randomly selected *clusters* from February 2013

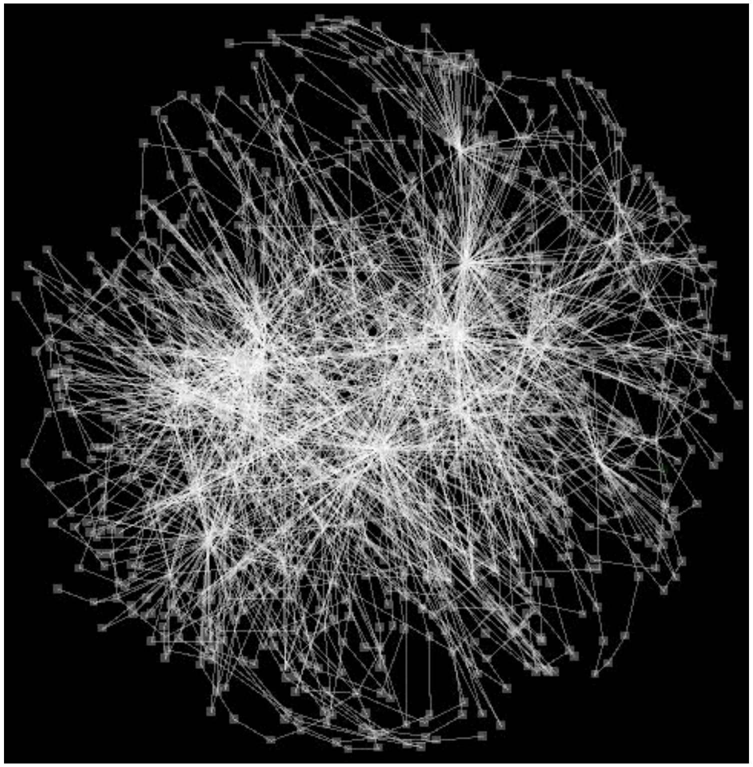
Roadmap



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

Carnegie Mellon

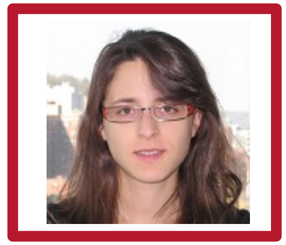
Wikipedia - editors



- Nodes: editors
- Edge A->B: ‘A’ changed ‘B’

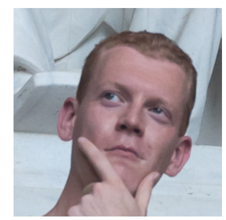
Any pattern?

VoG:Summarizing and Understanding Large Graphs



Danai Koutra,

U Kang,



Jilles Vreeken,

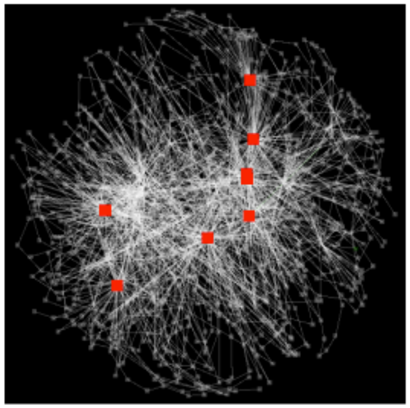
Christos Faloutsos.

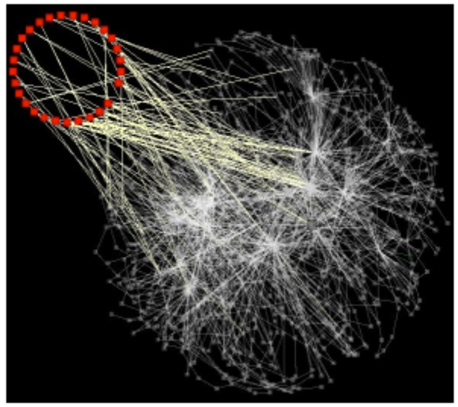
SDM 2014, Philadelphia, PA, April 2014.

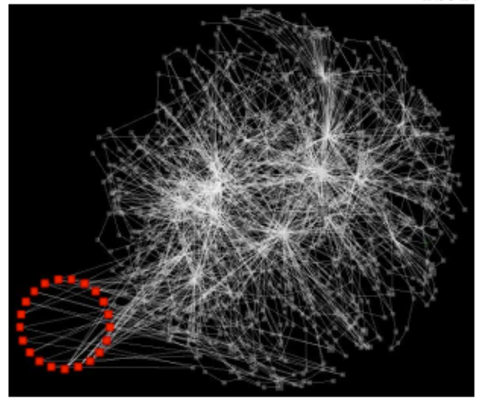
Code: `www.cs.cmu.edu/~dkoutra/CODE/vog.tar`

Carnegie Mellon

Wiki Controversial Article





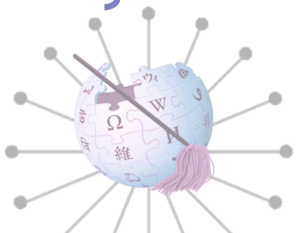


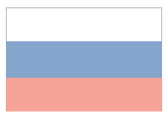
Stars: admins, bots, heavy users

Bipartite cores: edit wars

Kiev vs. Kyiv

vandals vs. admins





VoG: Summarizing and Understanding Large Graphs.
Koutra, Kang, Vreeken, Faloutsos. SDM '14.

VoG: Summarizing Graphs using Rich Vocabularies

Main Ideas:

(1) Use `vocabulary' of subgraph types

(2) Minimum Description Length (MDL) and above vocabulary, to summarize graph

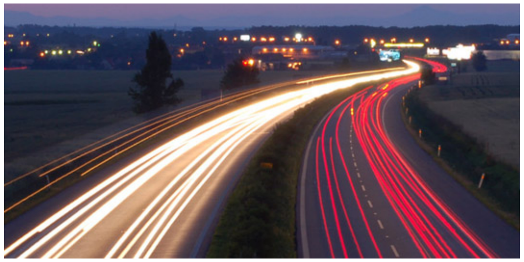
Carnegie Mellon

Summary of Part#1

- *many* patterns in real graphs
  - Power-laws everywhere
  - Gaussian trap
    - Avg << Max

Carnegie Mellon

Roadmap

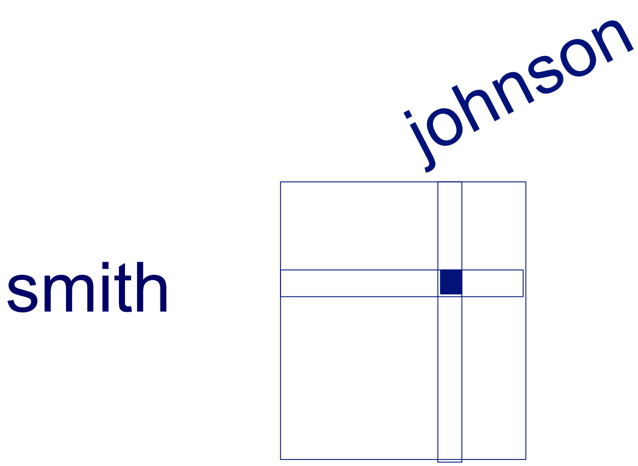


- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
  - P2.1: time-evolving graphs
  - P2.2: with side information ('coupled' M.T.F.)
  - Speed
- Part#3: Cascades and immunization
- Conclusions

Part 2: Time evolving graphs; tensors

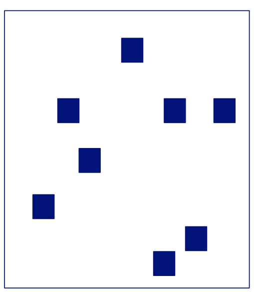
Graphs over time -> tensors!

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



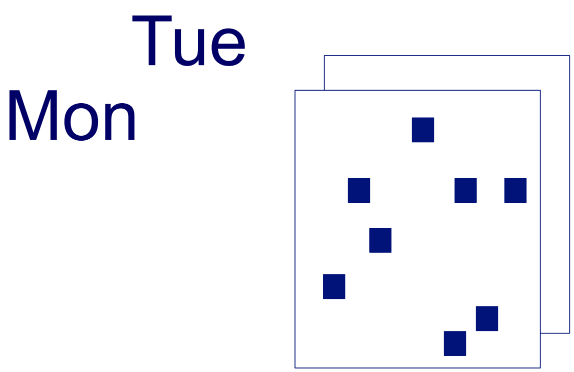
Graphs over time -> tensors!

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



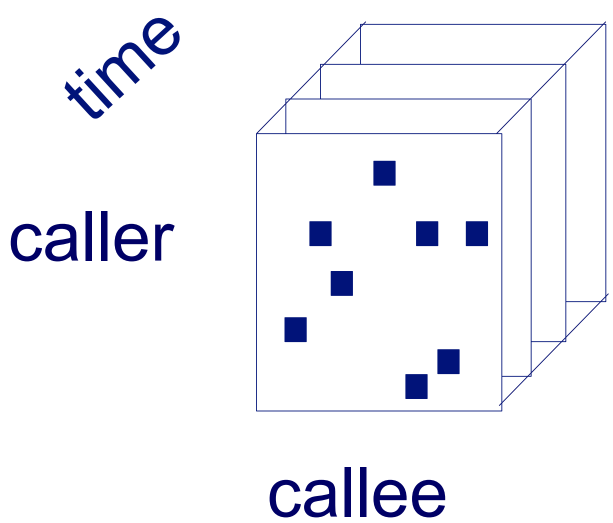
Graphs over time -> tensors!

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



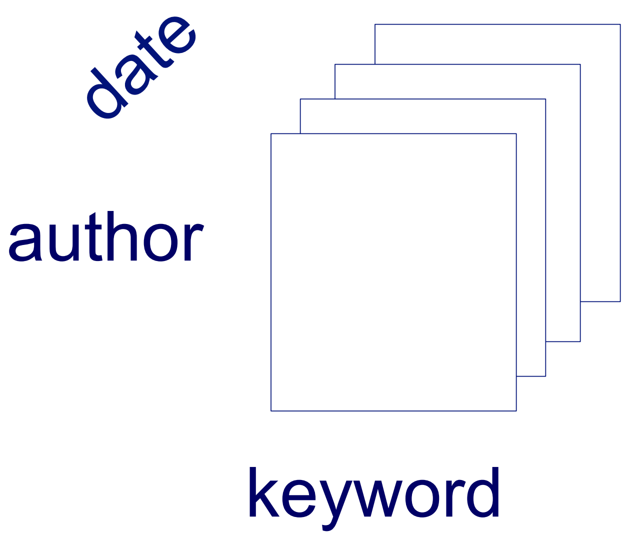
Graphs over time -> tensors!

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



Graphs over time -> tensors!

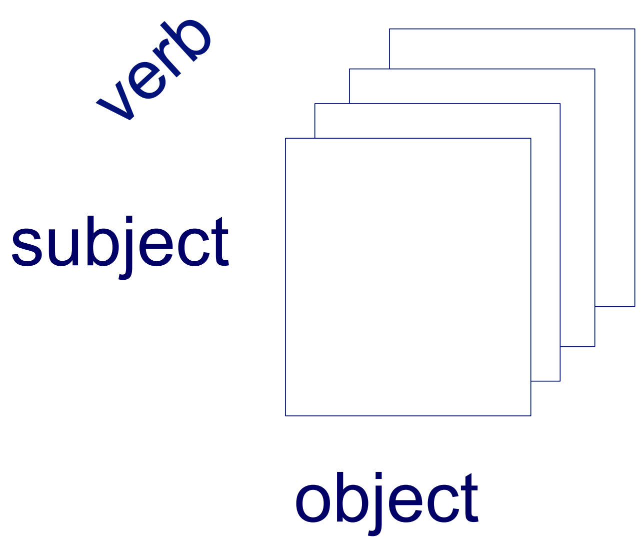
- Problem #2.1':
  - Given author-keyword-date
  - Find patterns / anomalies



MANY more settings, with >2 'modes'

Graphs over time -> tensors!

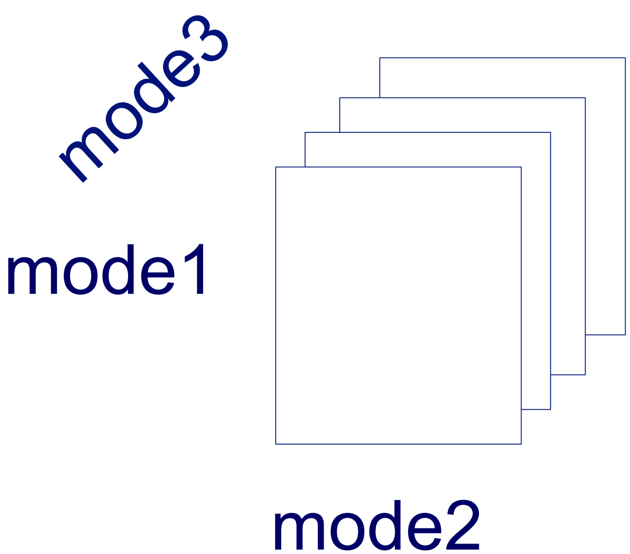
- Problem #2.1'':
  - Given subject – verb – object facts
  - Find patterns / anomalies



MANY more settings, with >2 'modes'

Graphs over time -> tensors!

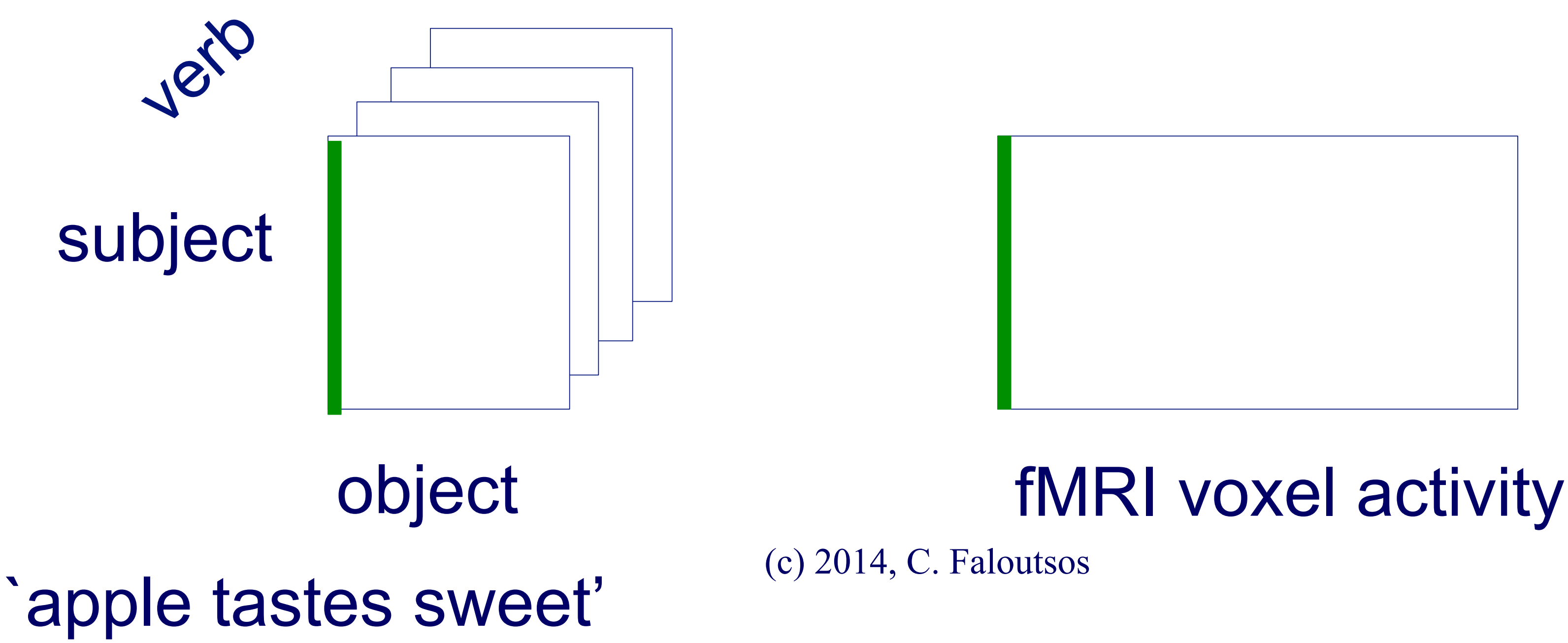
- Problem #2.1''':
  - Given <triplets>
  - Find patterns / anomalies



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

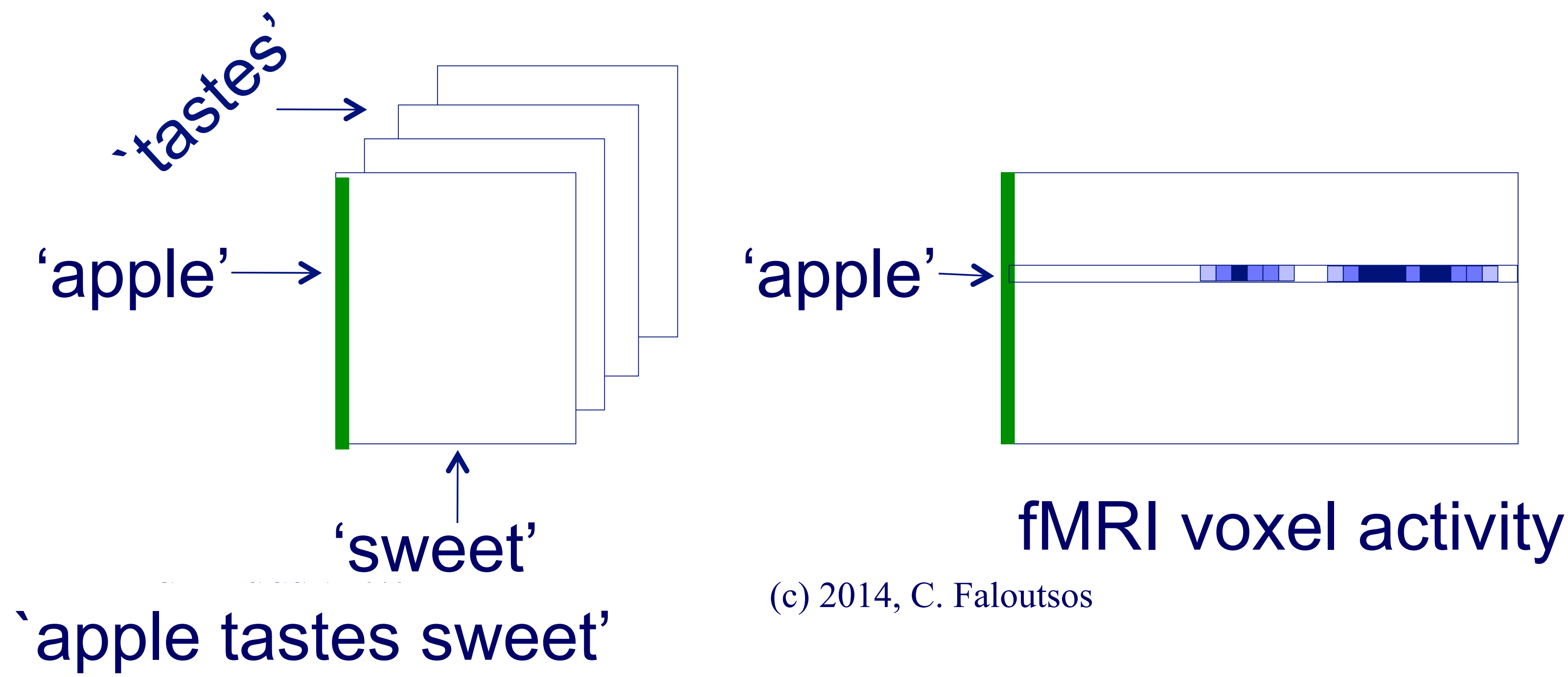
Graphs & side info

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



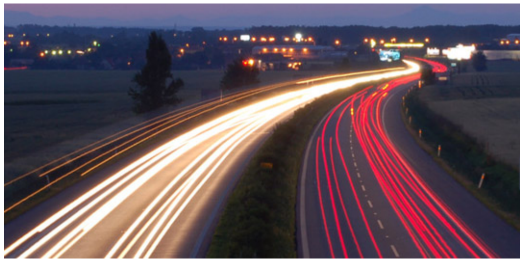
Graphs & side info

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



Carnegie Mellon

Roadmap

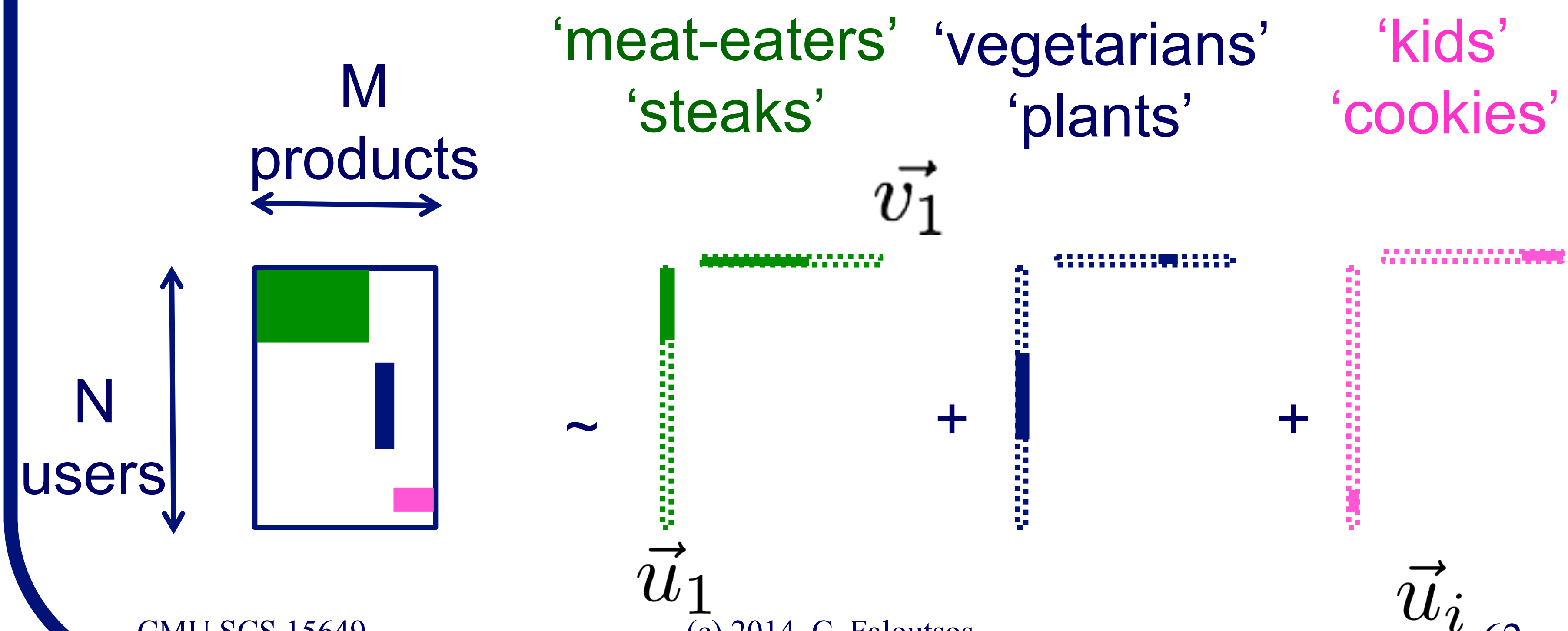


- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
  - P2.1: time-evolving graphs
  - P2.2: with side information (‘coupled’ M.T.F.)
  - Speed
- Part#3: Cascades and immunization
- Conclusions

Carnegie Mellon

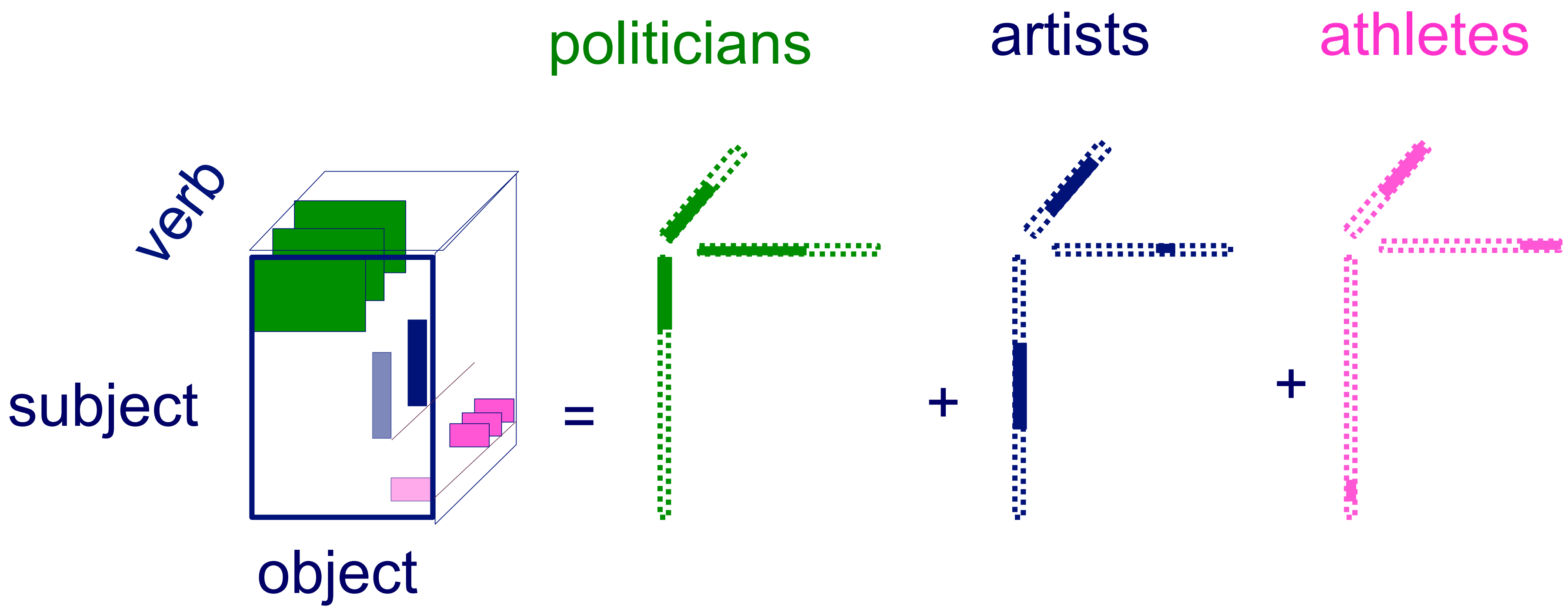
Answer to both: tensor factorization

- Recall: (SVD) matrix factorization: finds blocks



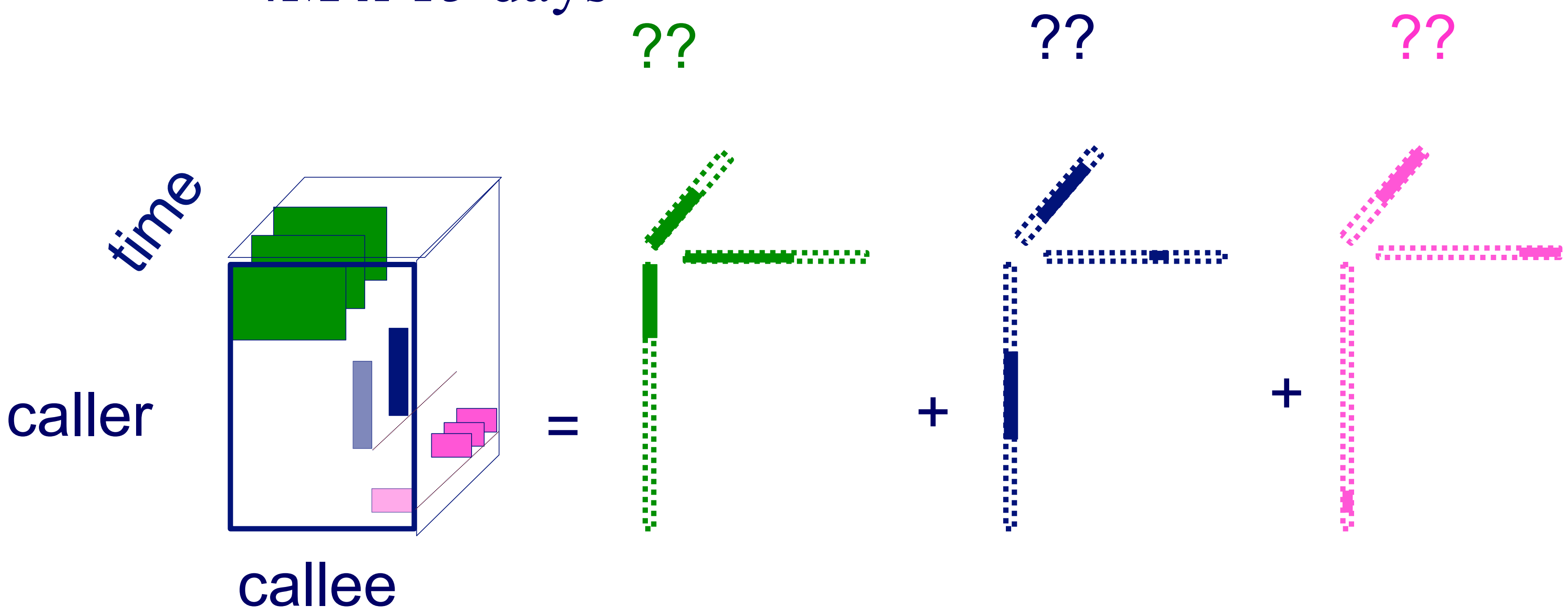
Answer to both: tensor factorization

- PARAFAC decomposition

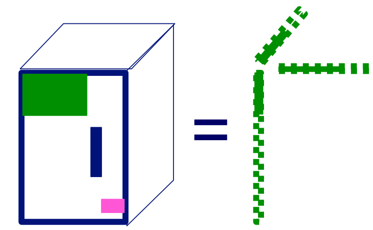


Answer: tensor factorization

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

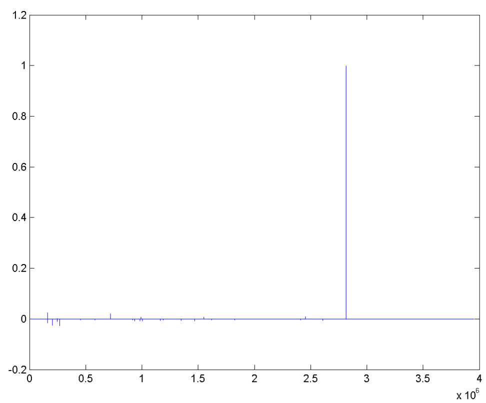


Anomaly detection in time-evolving graphs

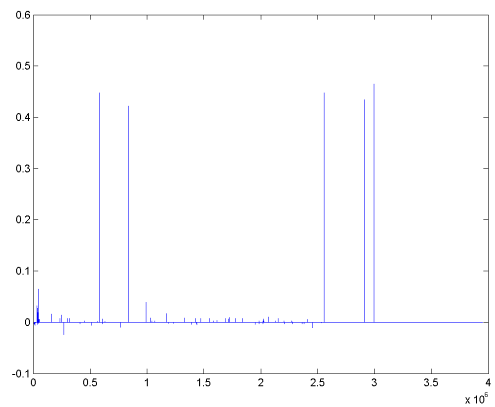


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

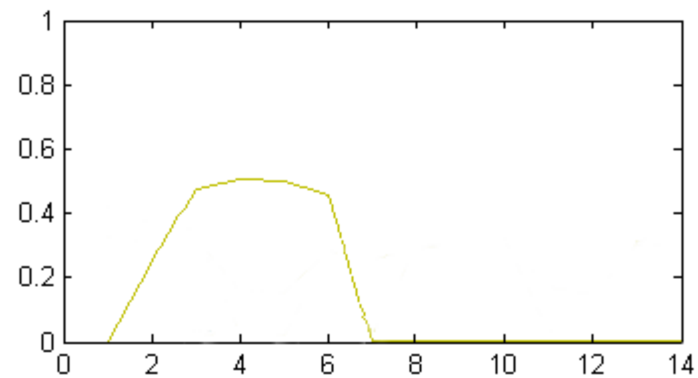
1 caller



5 receivers

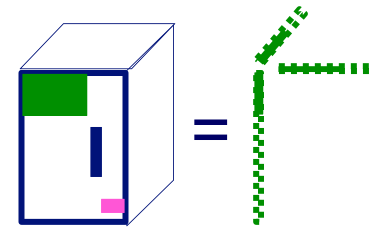


4 days of activity



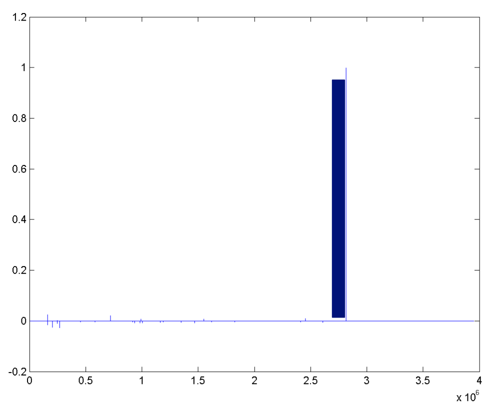
~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs

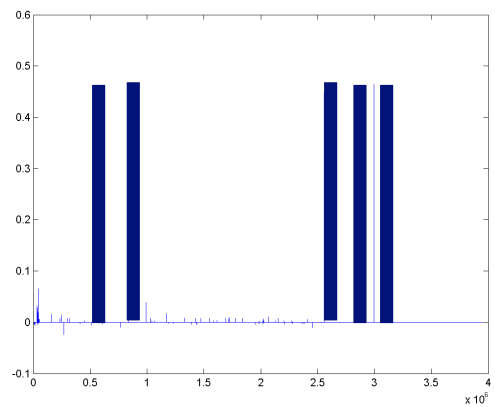


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

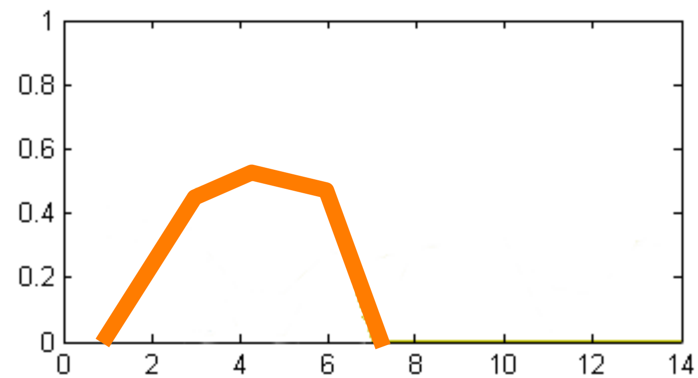
1 caller



5 receivers



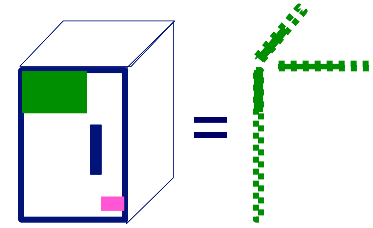
4 days of activity



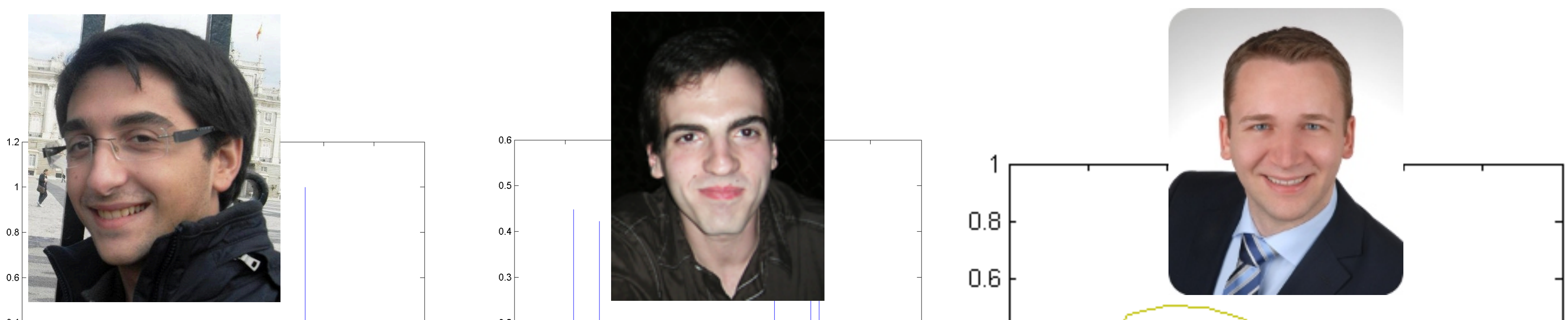
~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs



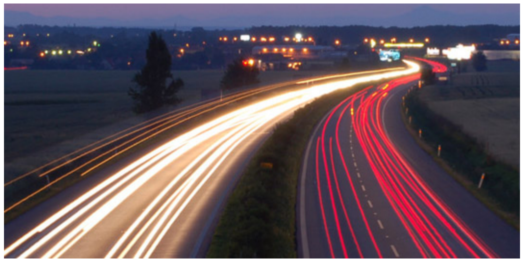
- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks



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.

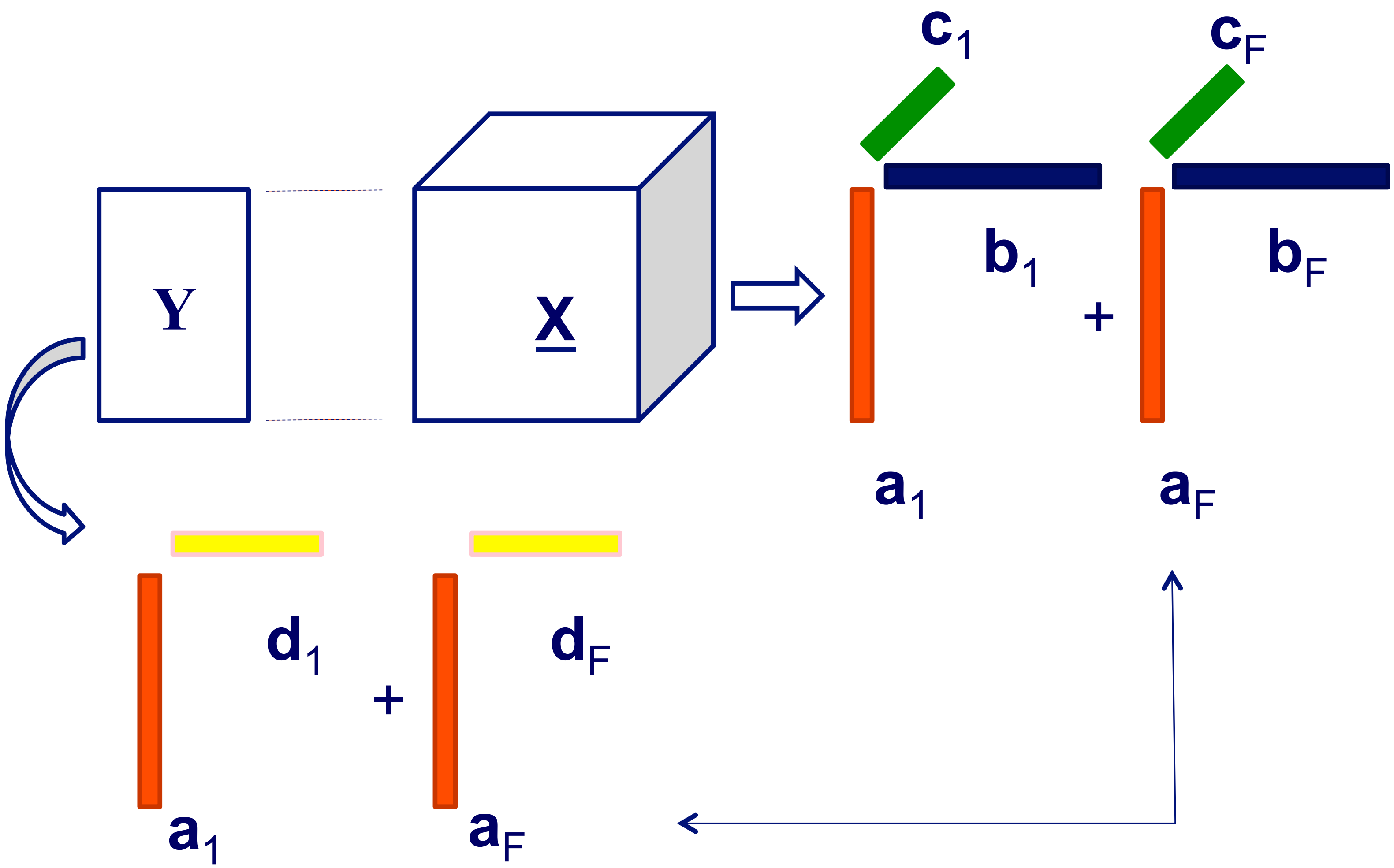
Carnegie Mellon

Roadmap



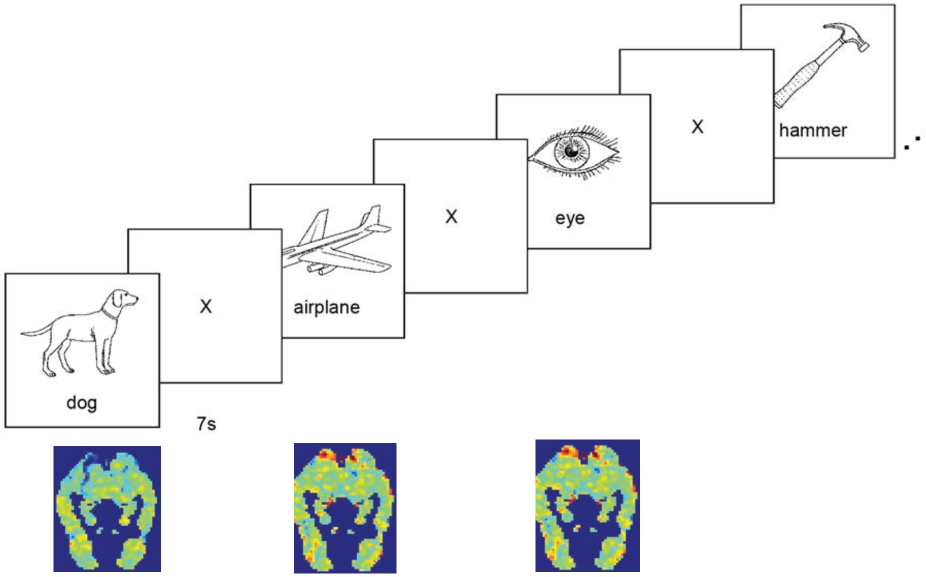
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
  - P2.1: Discoveries @ phonecall network
  - P2.2: Discoveries in neuro-semantics
  - Speed
- Part#3: Cascades and immunization
- Conclusions

Coupled Matrix-Tensor Factorization (CMTF)



Neuro-semantics

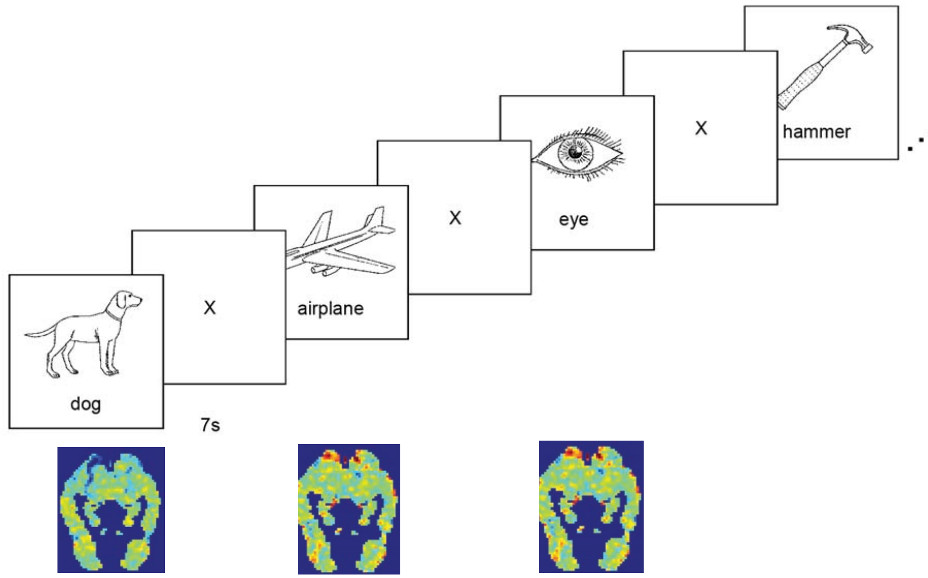
- **Brain Scan Data***
  - 9 persons
  - 60 nouns
- **Questions**
  - 218 questions
  - 'is it alive?', 'can you eat it?'



*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-semantics

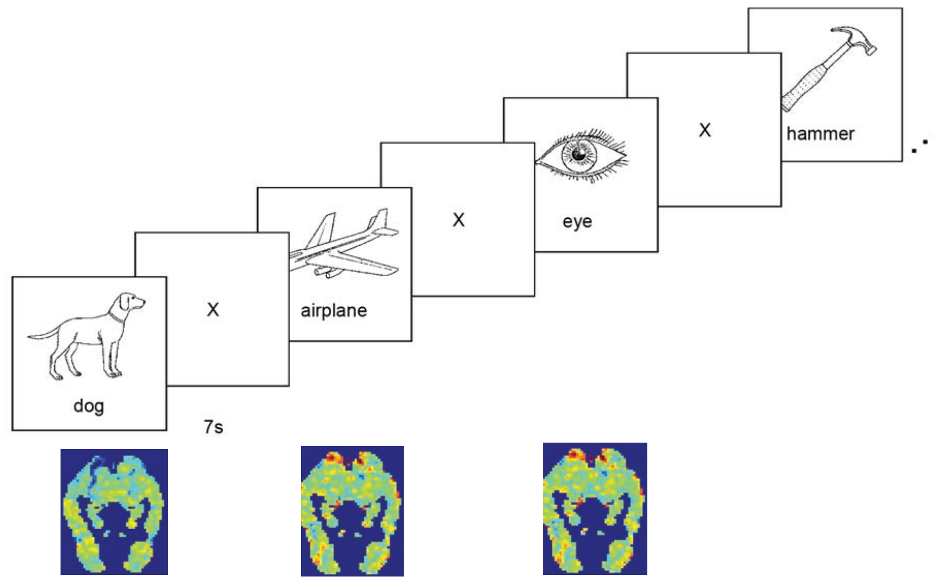
- **Brain Scan Data***
  - 9 persons
  - 60 nouns
- **Questions**
  - 218 questions
  - 'is it alive?', 'can you eat it?'



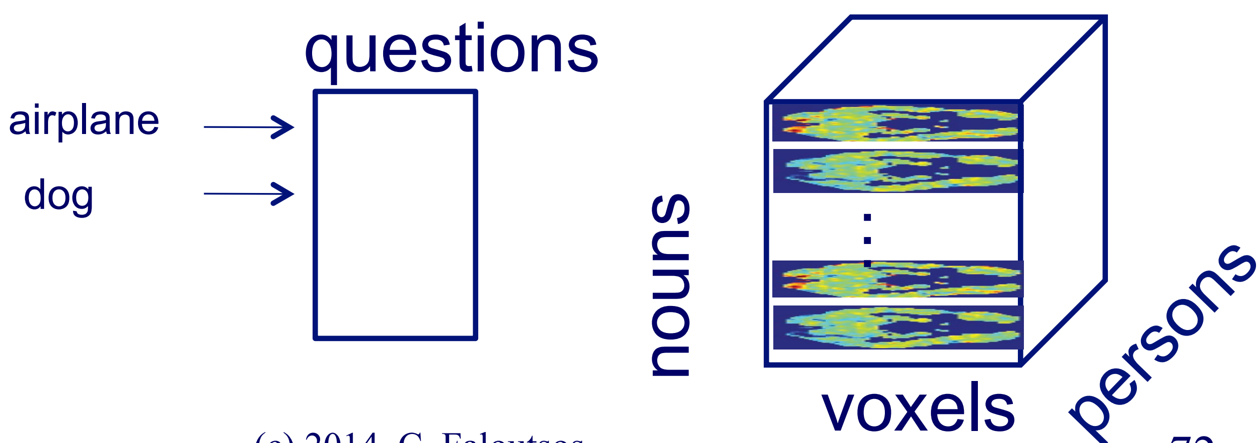
Patterns?

Neuro-semantics

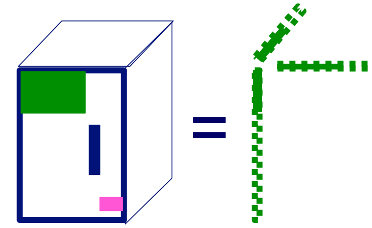
- **Brain Scan Data***
  - 9 persons
  - 60 nouns
- **Questions**
  - 218 questions
  - 'is it alive?', 'can you eat it?'



Patterns?



Neuro-semantics



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

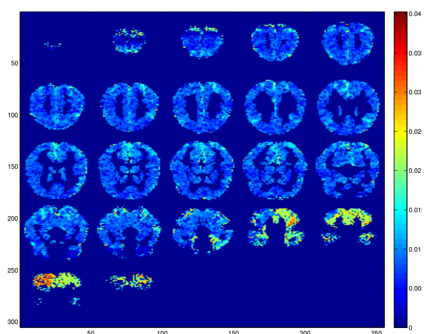
can you pick it up?
can you hold it in one hand?
is it smaller than a golfball?

Nouns

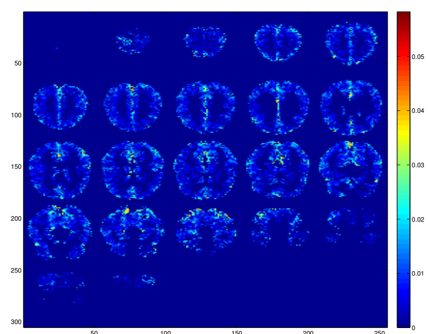
bed
house
car

Questions

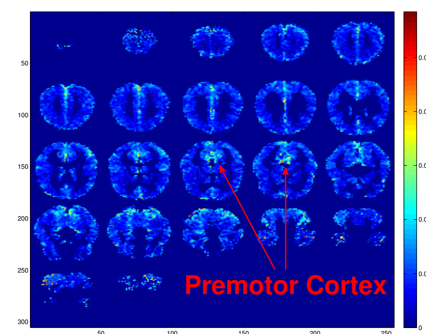
does it use electricity?
can you sit on it?
does it cast a shadow?



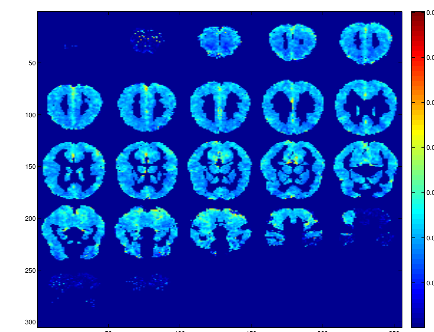
Group1



Group 2

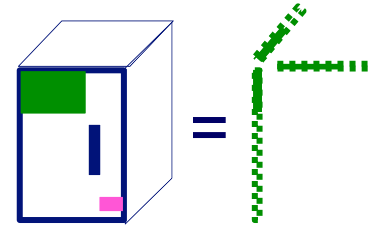


Group 3



Group 4

Neuro-semantics



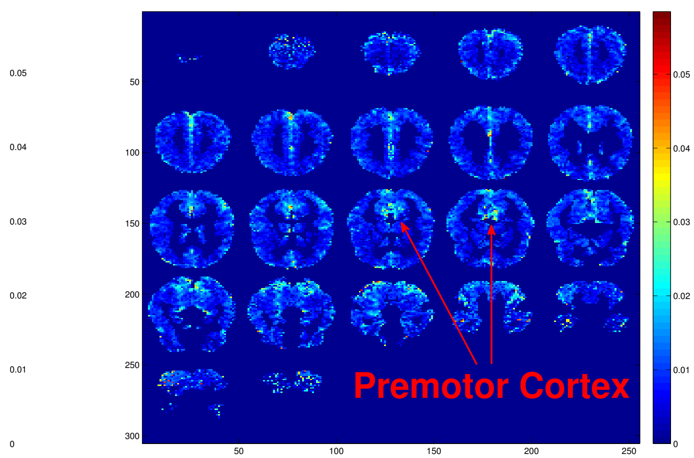
Small items -> Premotor cortex

Nouns

glass
tomato
bell

Questions

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



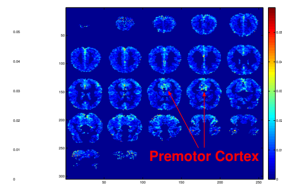
Group 3

Neuro-semantics

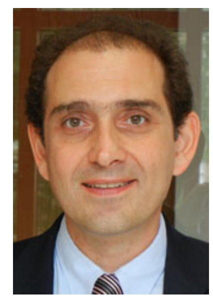
Small items -> Premotor cortex

Nouns
glass
tomato
bell

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

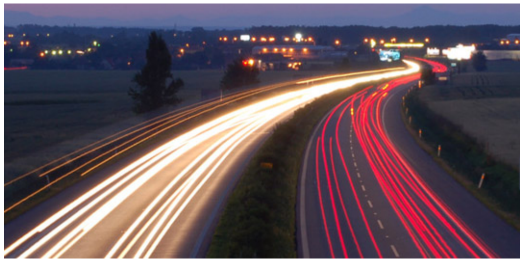


Group 3



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
  - Speed
- Part#3: Cascades and immunization
- Conclusions

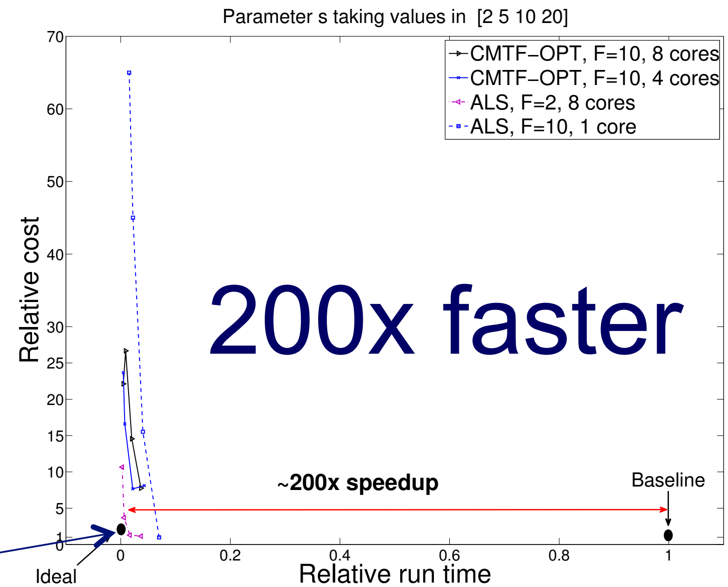
Speed of tensor/CMTF analysis

- Q1: Can we make it fast?
- Q2: Does it work for large, disk-based data?

Carnegie Mellon

A1: Turbo-SMT

Relative cost

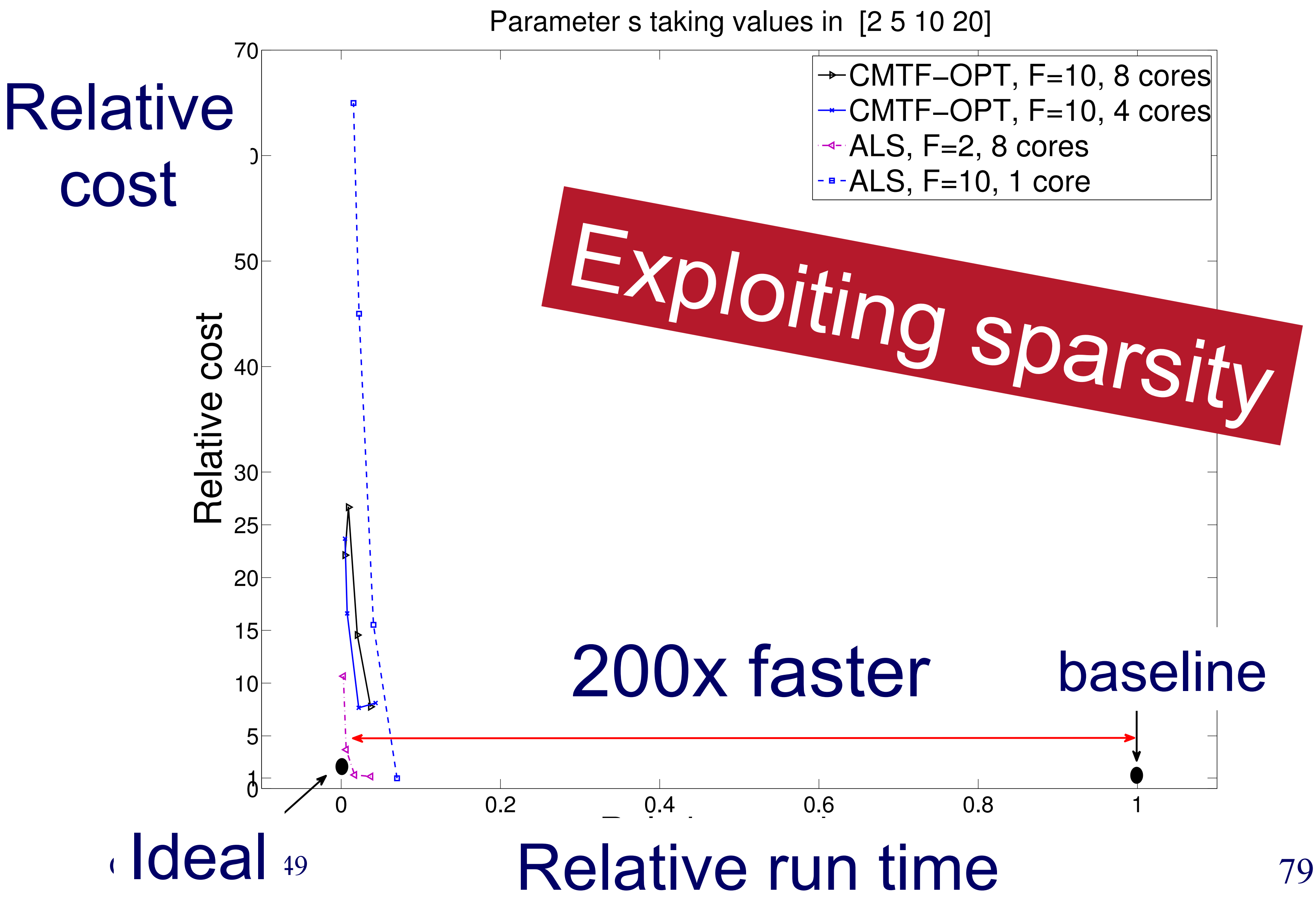


Ideal

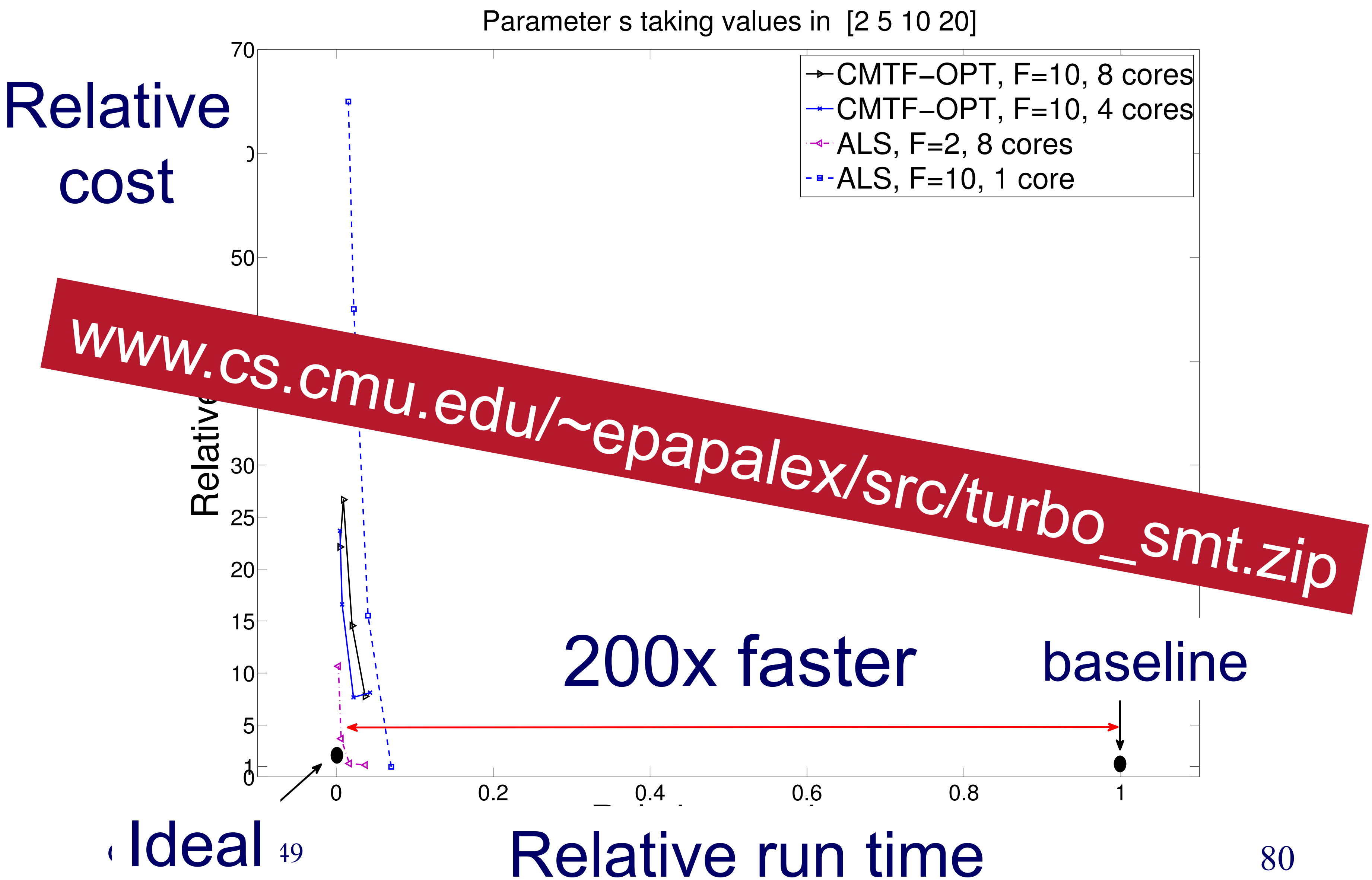
Relative run time

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

A1: Turbo-SMT


Parameter s taking values in [2 5 10 20]
CMTF−OPT, F=10, 8 cores
CMTF−OPT, F=10, 4 cores
ALS, F=2, 8 cores
ALS, F=10, 1 core
Relative cost
70
50
40
30
25
20
15
10
5
1
0
0
0.2
0.4
0.6
0.8
1
Exploiting sparsity
200x faster
baseline
Relative
cost
Ideal
Relative run time

A1: Turbo-SMT


Parameter s taking values in [2 5 10 20]
Relative cost
CMTF−OPT, F=10, 8 cores
CMTF−OPT, F=10, 4 cores
ALS, F=2, 8 cores
ALS, F=10, 1 core
www.cs.cmu.edu/~epapalex/src/turbo_smt.zip
200x faster
baseline
Ideal
Relative run time
70
50
30
25
20
15
10
5
1
0
0
0.2
0.4
0.6
0.8
1

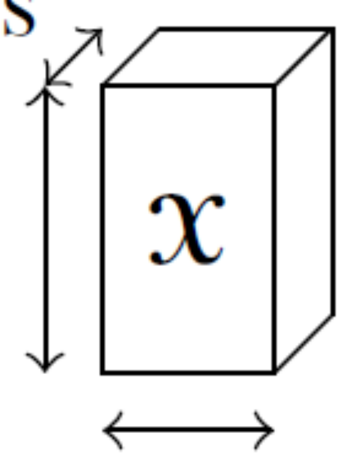
Q2: spilling to the disk?

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

144M non-zeros

48M verbs

26M subjects



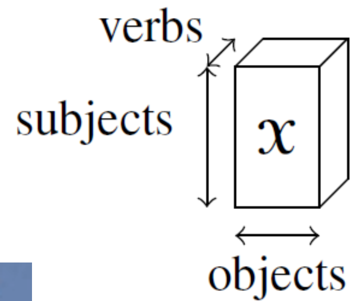
26M objects

NELL (Never Ending Language Learner) @CMU

A2: GigaTensor

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

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

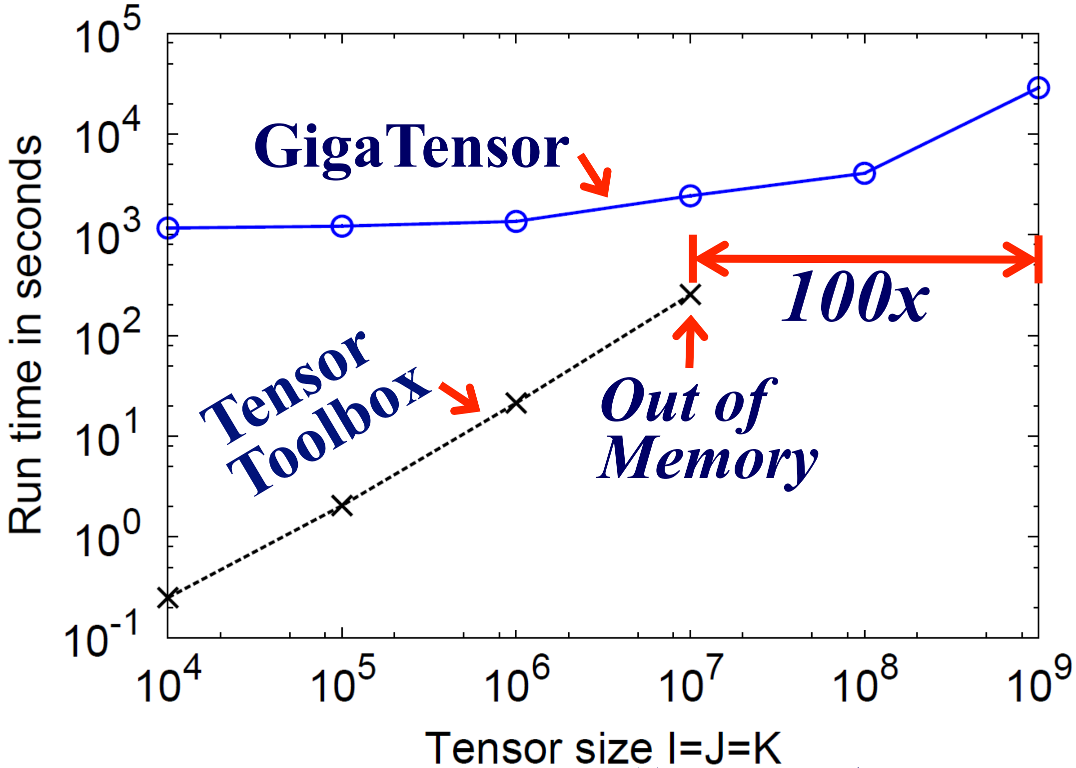


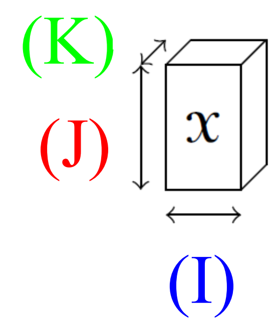
NELL (Never Ending Language Learner) @CMU

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

A2: GigaTensor

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

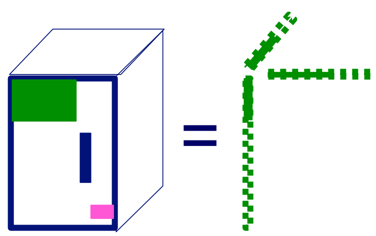




Number of nonzero = I / 50

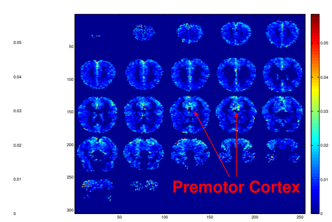
Part 2: Conclusions

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



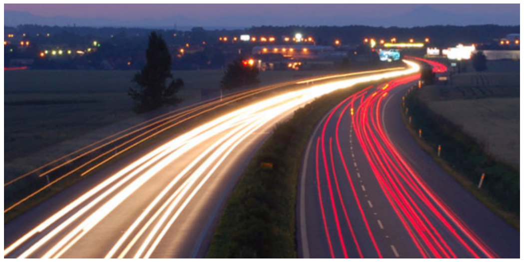
Nouns
glass
tomato
bell

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



Group 3

Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Conclusions

Project info: PEGASUS

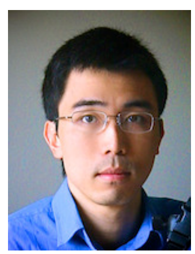
www.cs.cmu.edu/~pegasus

Results on large graphs: with Pegasus + hadoop + M45

Apache license

Code, papers, manual, video

Prof. U Kang



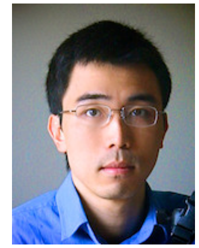
Prof. Polo Chau

Cast

Akoglu, Leman

Araujo, Miguel

Beutel, Alex



Chau, Polo

Kang, U

Koutra, Danai

Lee, Jay Yoon

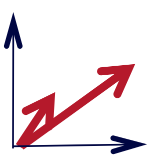
Prakash, Aditya

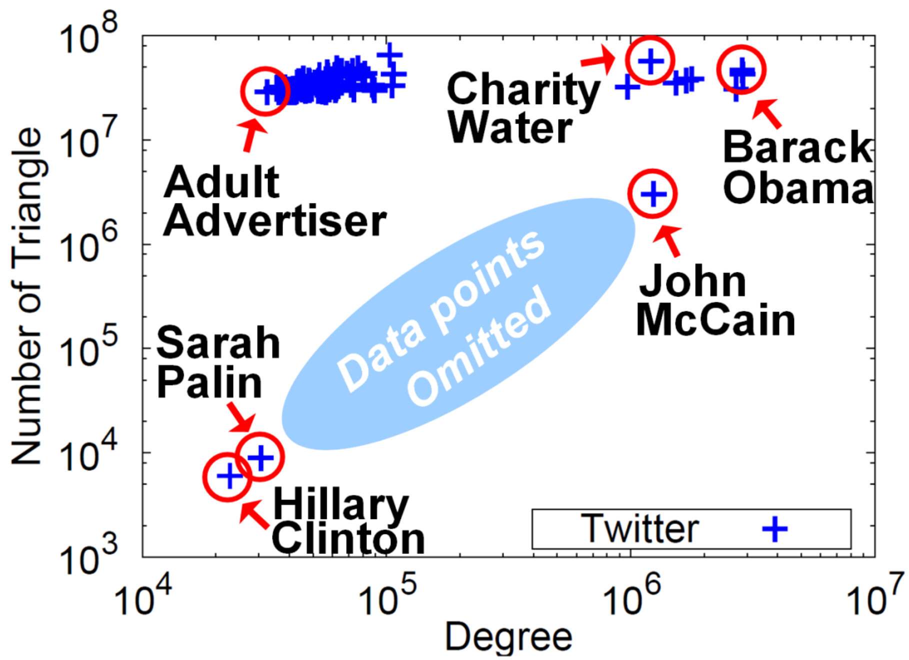
Papalexakis, Vagelis

Shah, Neil

CONCLUSION#1 – Big data

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



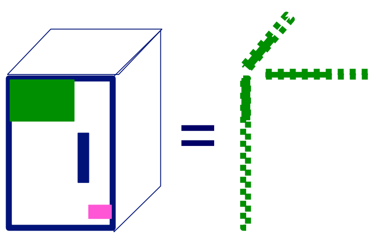


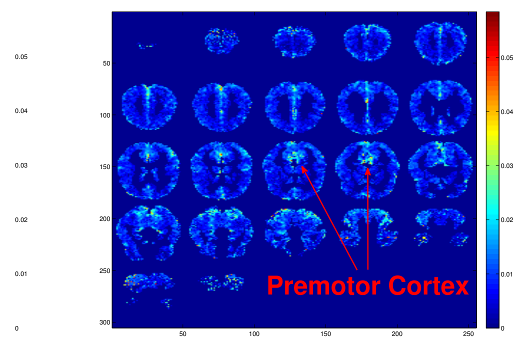
CONCLUSION#2 – tensors

- 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?’

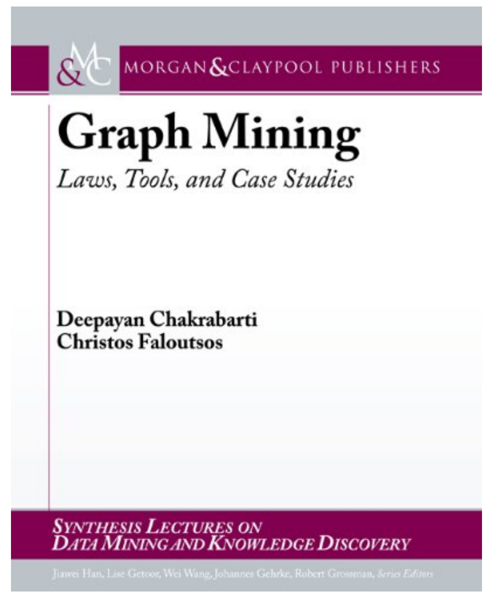




Group 3

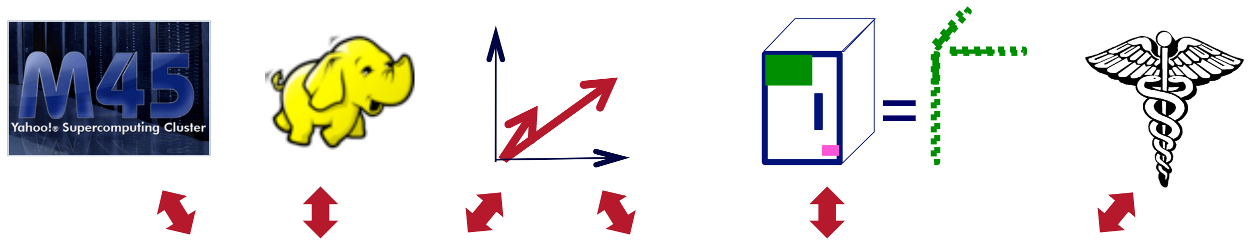
References

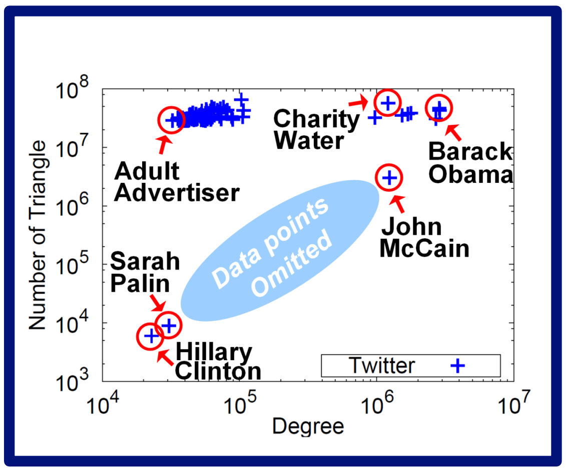
- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
- http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006

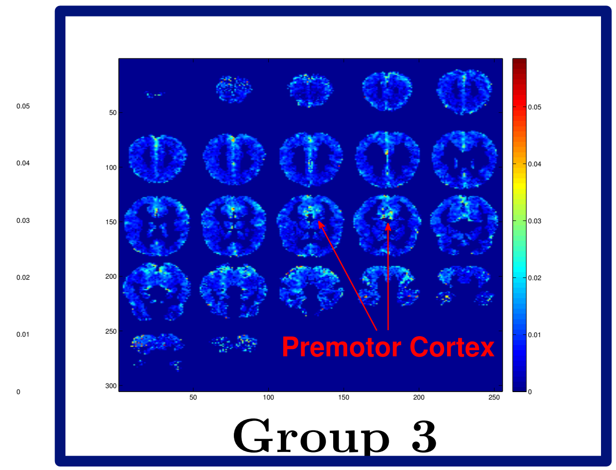


TAKE HOME MESSAGE:

Cross-disciplinarity







Questions?

Cross-disciplinarity