Carnegie Mellon

Mining Large Graphs and Time Sequences: Patterns, Anomalies, and Fraud Detection

Christos Faloutsos

CMU

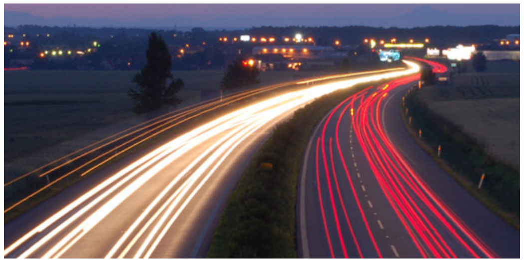
Carnegie Mellon

Thank you!

- Alkis Polyzotis

- Denise Olivera

Roadmap



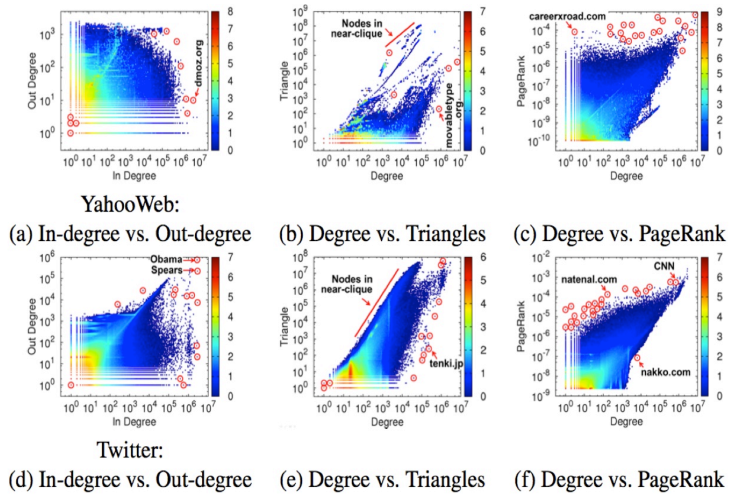
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Part#3: time sequences
- Conclusions

Graphs - why should we care?

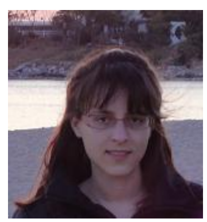
>$10B; ~1B users

Carnegie Mellon

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.

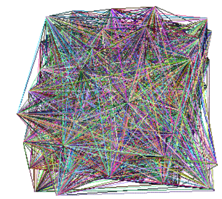
Carnegie Mellon

Graphs - why should we care?

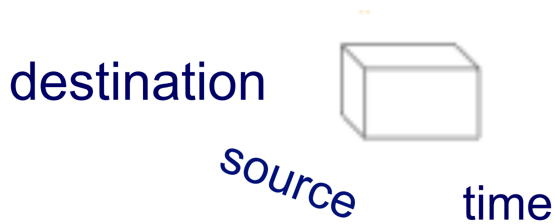
- web-log ('blog') news propagation YAHOO! BLOG
- computer network security: email/IP traffic and anomaly detection
- Recommendation systems NETFLIX
- ....

- Many-to-many db relationship -> graph

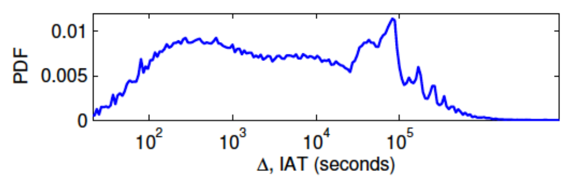
Motivating problems



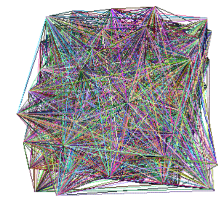
- P1: patterns? Fraud detection?
- P2: patterns in time-evolving graphs / tensors



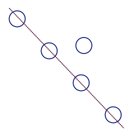
- P3: time sequences



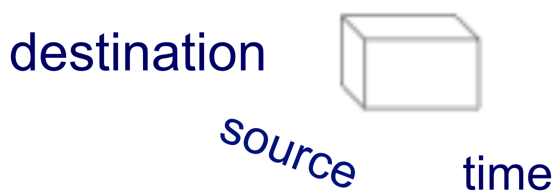
Motivating problems



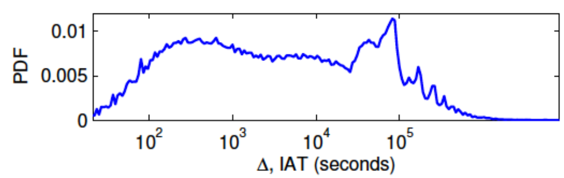
- P1: patterns? Fraud detection?

 Patterns  anomalies

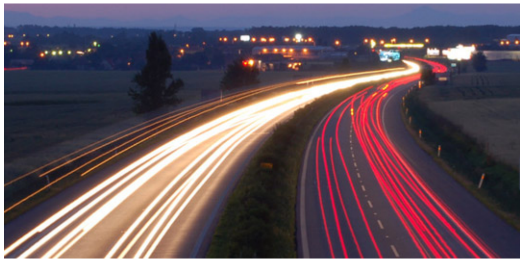
- P2: patterns in time-evolving graphs / tensors



- P3: time sequences

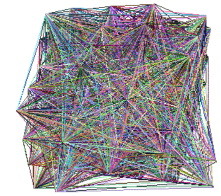


Roadmap



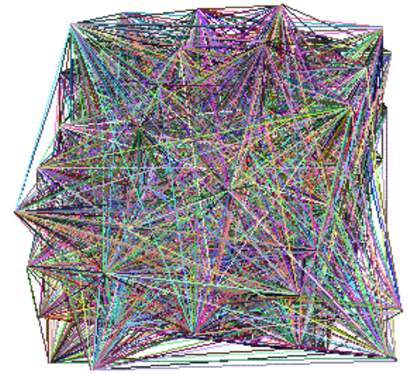
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns & fraud detection
- Part#2: time-evolving graphs; tensors
- Conclusions

Carnegie Mellon



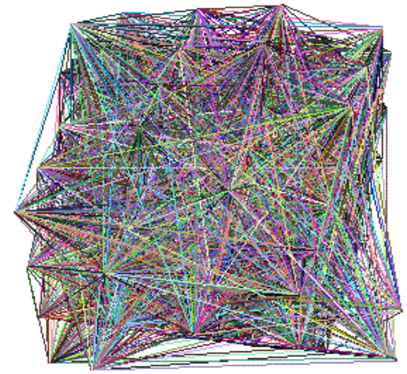
Part 1: Patterns, & fraud detection

Laws and patterns

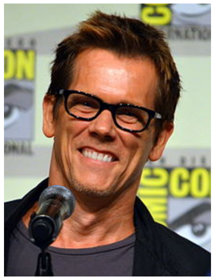


- Q1: Are real graphs random?

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

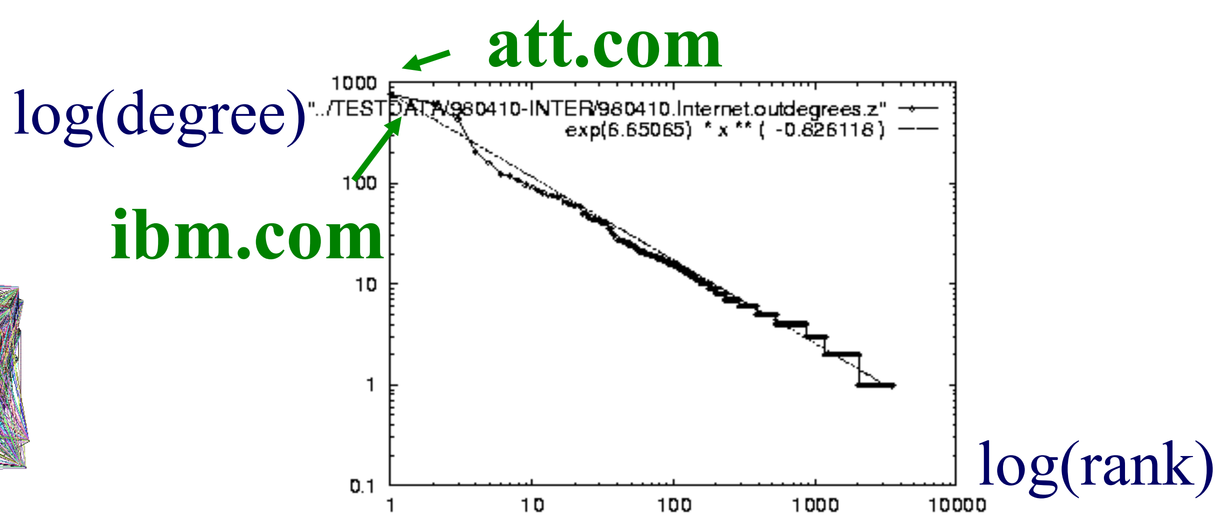


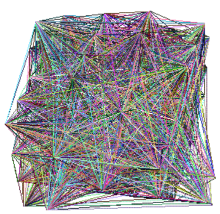
Carnegie Mellon

Solution# S.1

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

internet domains

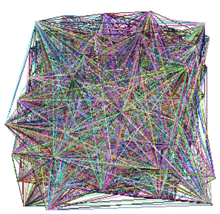




Solution# S.1

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

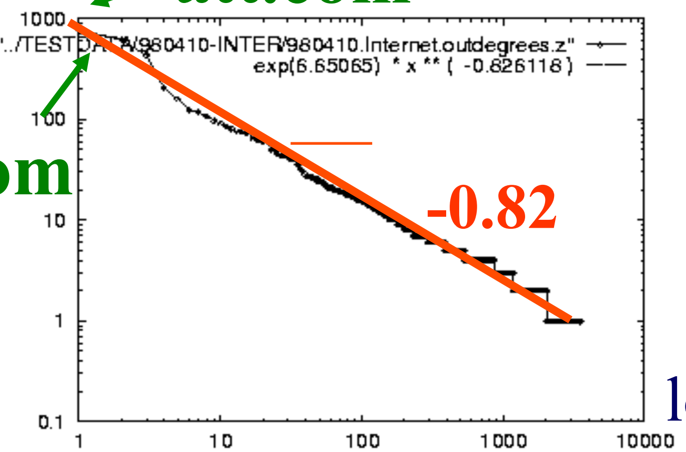
internet domains



att.com

log(degree)

ibm.com



-0.82

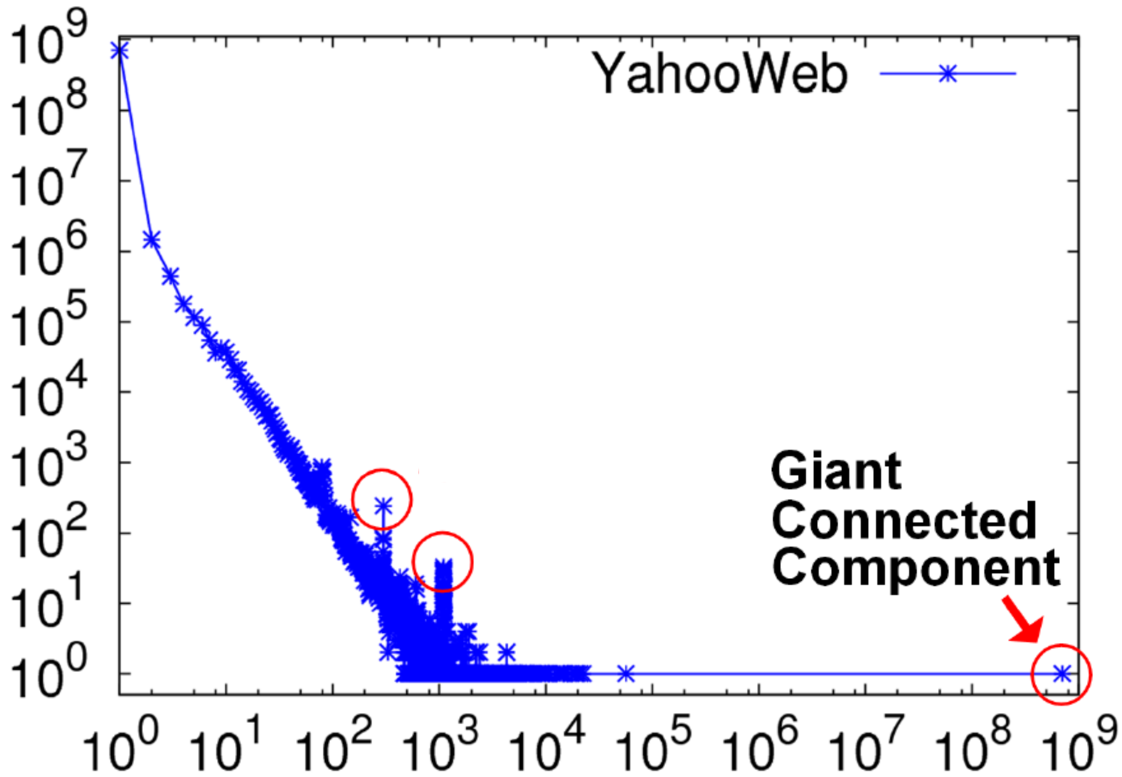
log(rank)

Carnegie Mellon

S2: connected component sizes

- Connected Components – 4 observations:

Count

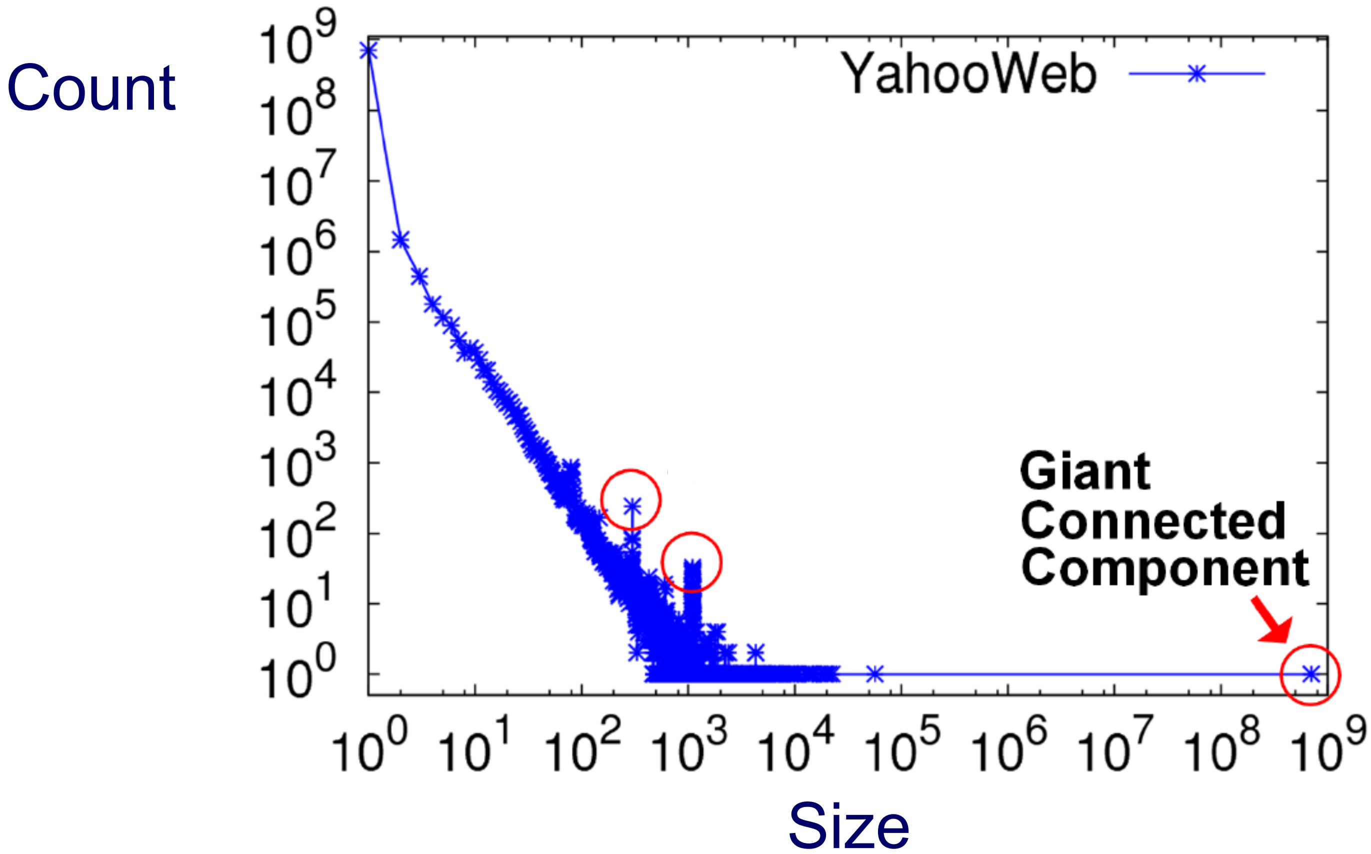


Size

1.4B nodes
6B edges

S2: connected component sizes

- Connected Components



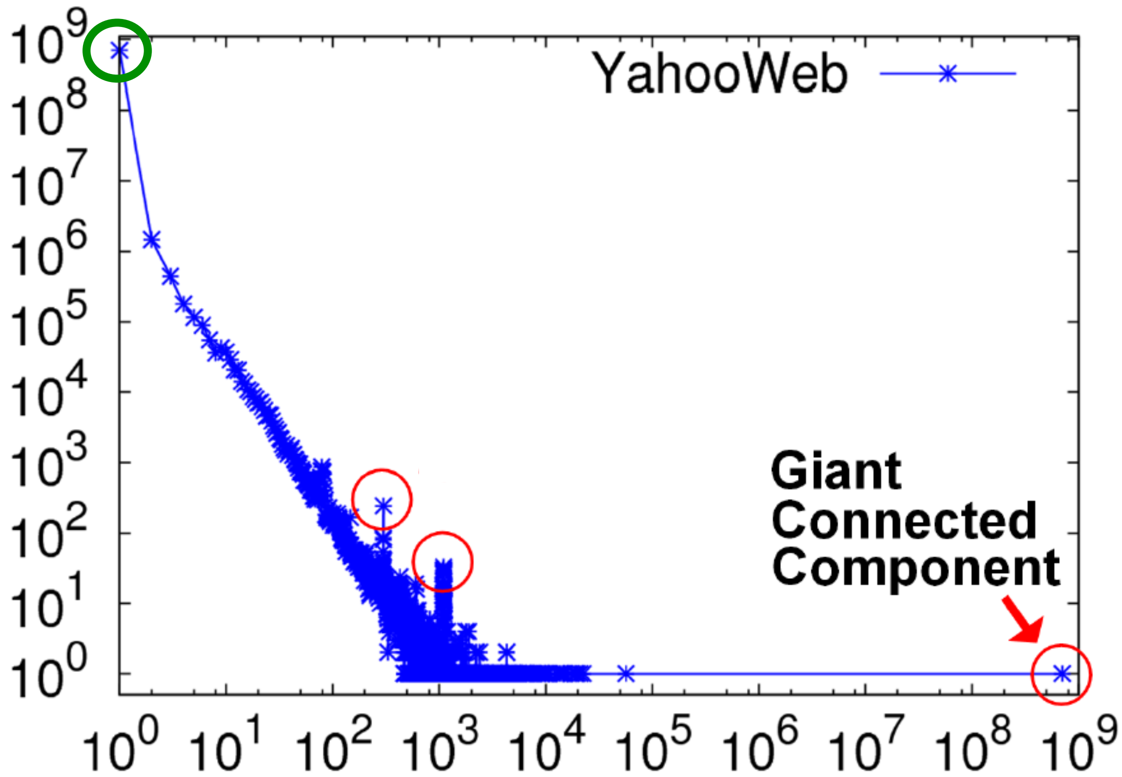
1) 10K x larger than next

S2: connected component sizes

- Connected Components

Count

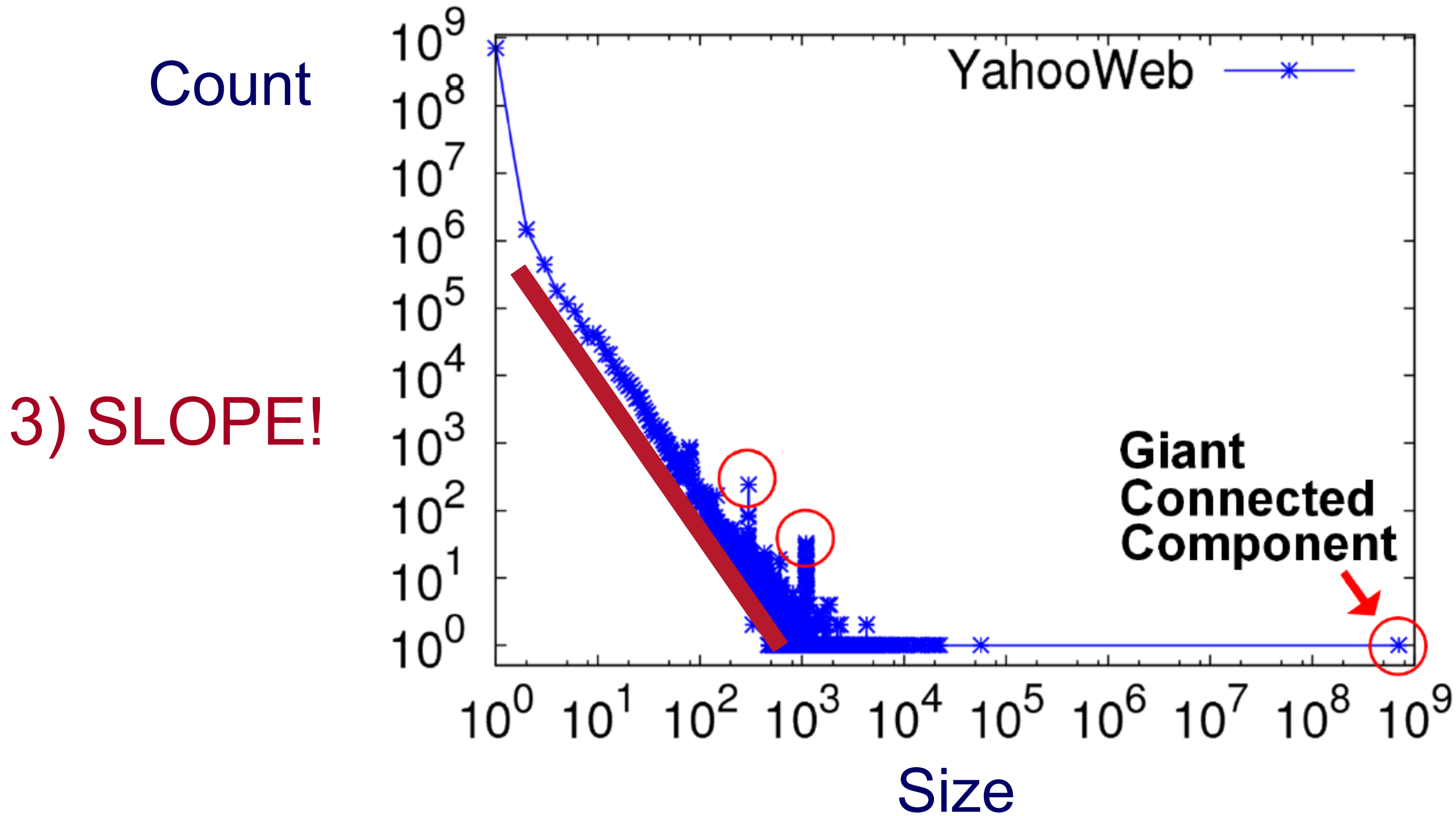
2) ~0.7B singleton nodes



Size

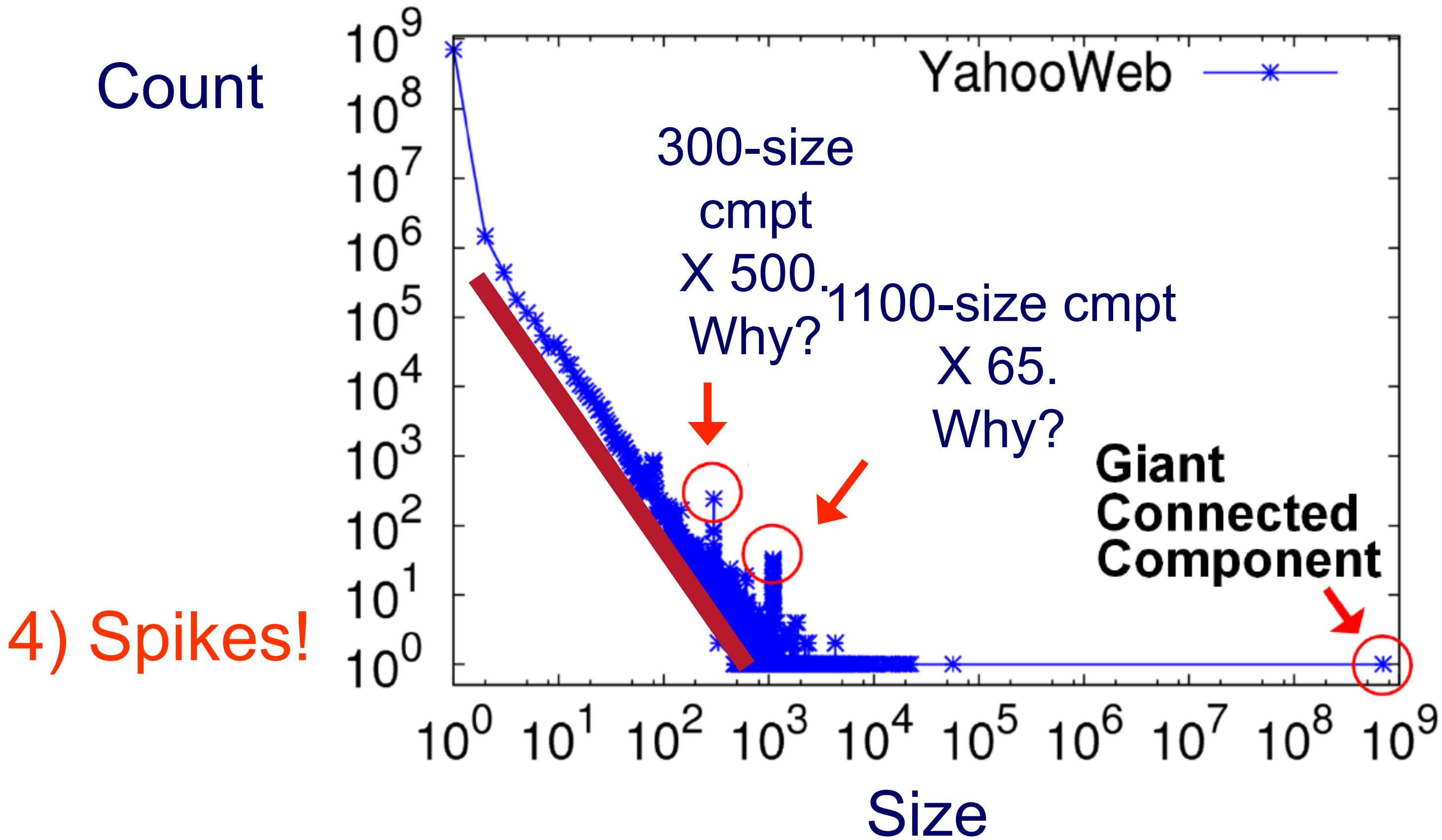
S2: connected component sizes

- Connected Components



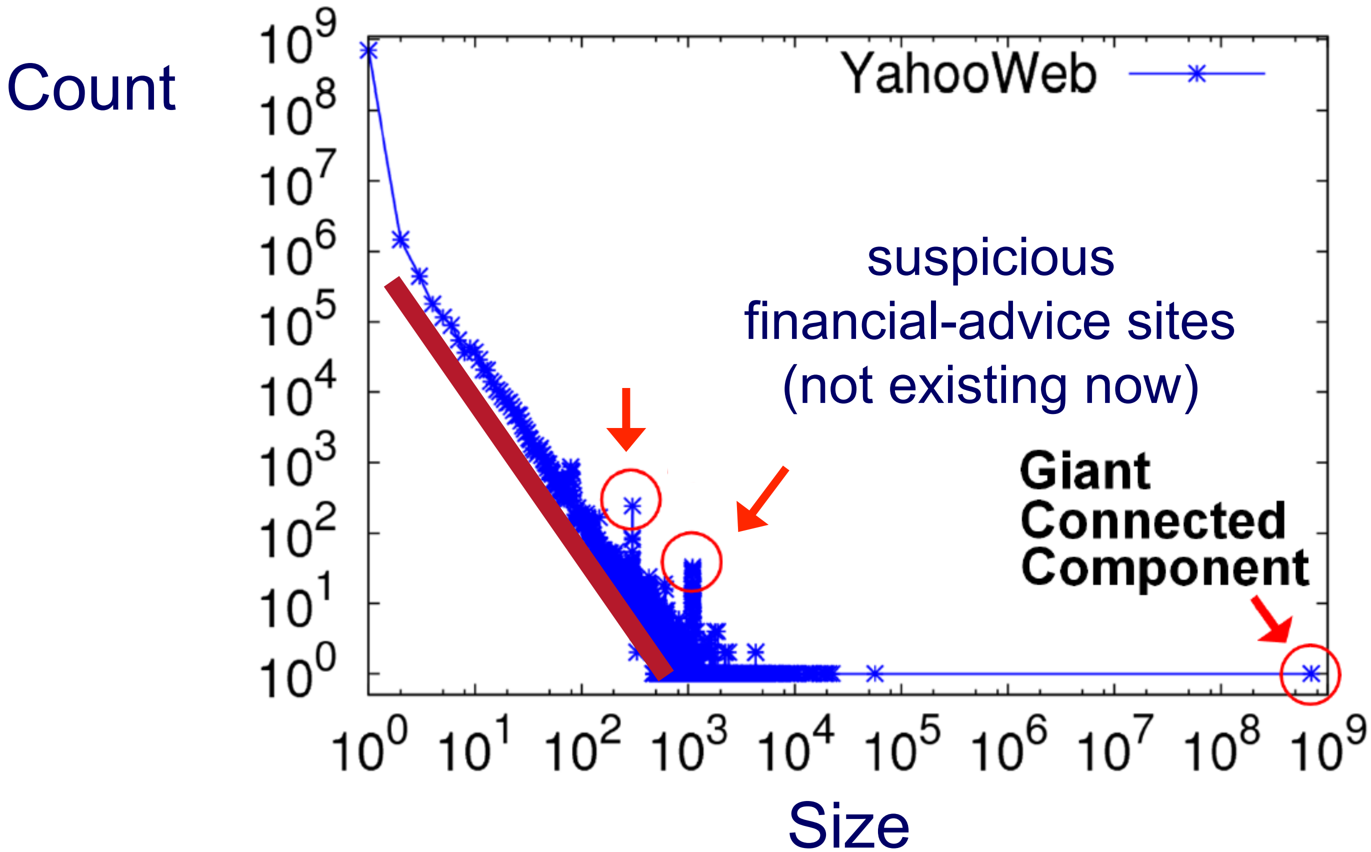
S2: connected component sizes

- Connected Components



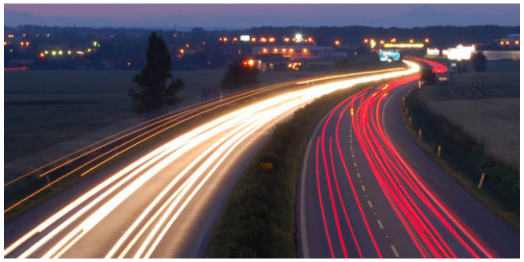
S2: connected component sizes

- Connected Components



Carnegie Mellon

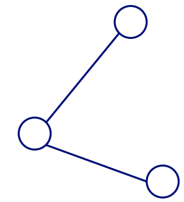
Roadmap



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

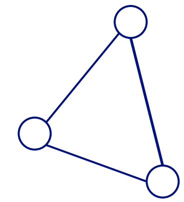
Carnegie Mellon

Solution# S.3: Triangle 'Laws'

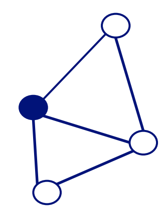


- Real social networks have a lot of triangles

Solution# S.3: Triangle 'Laws'

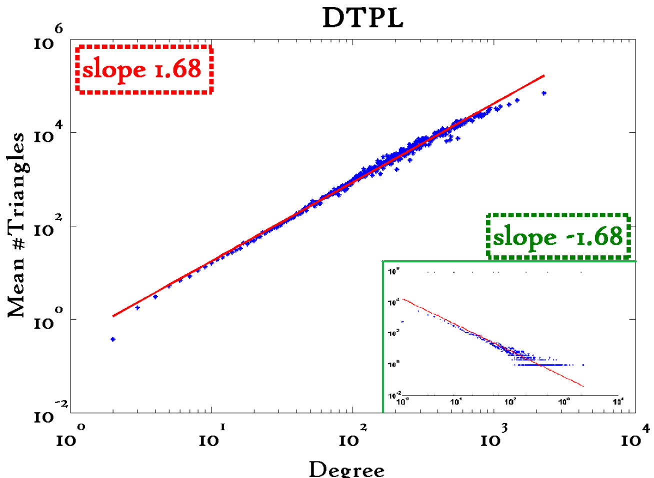


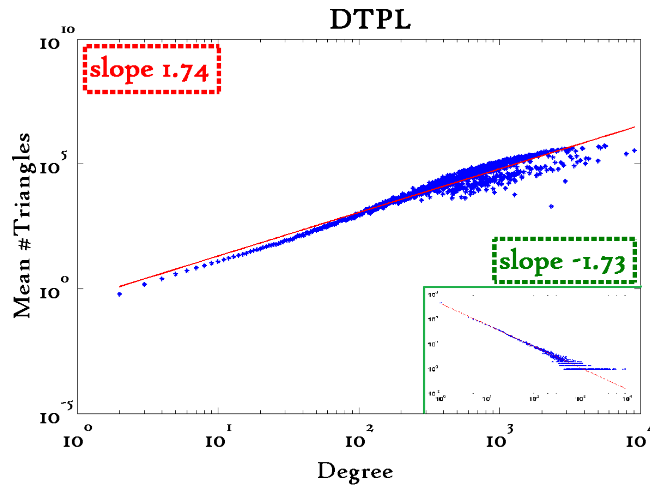
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?



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

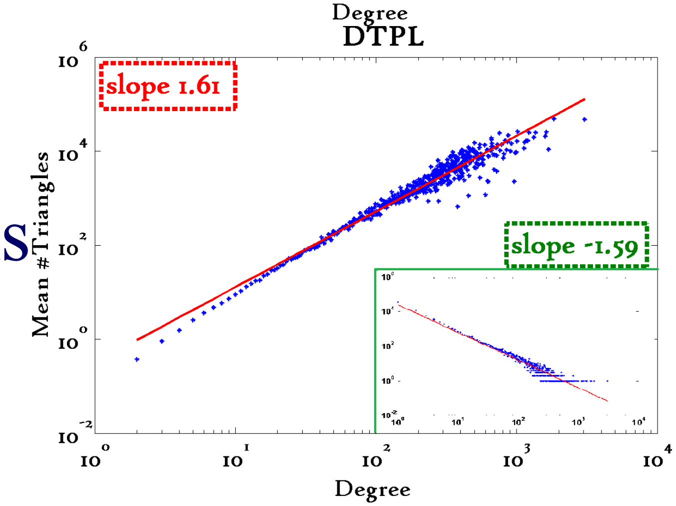
Reuters

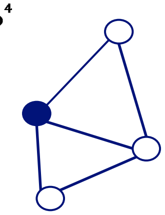




SN

Epinions



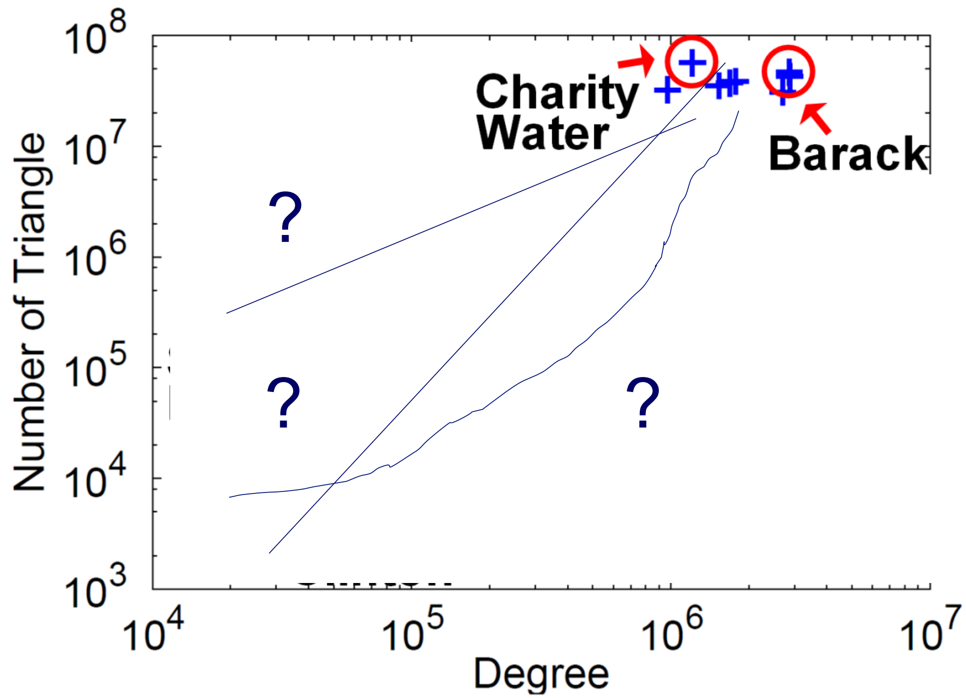


X-axis: degree

Y-axis: mean # triangles

n friends -> ~$n^{1.6}$ triangles

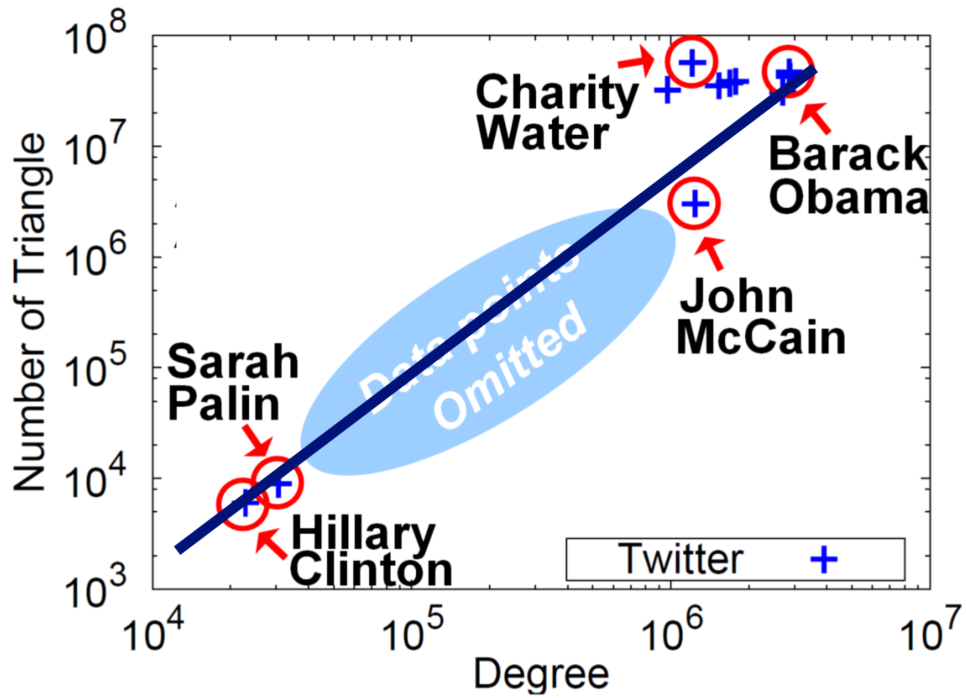
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

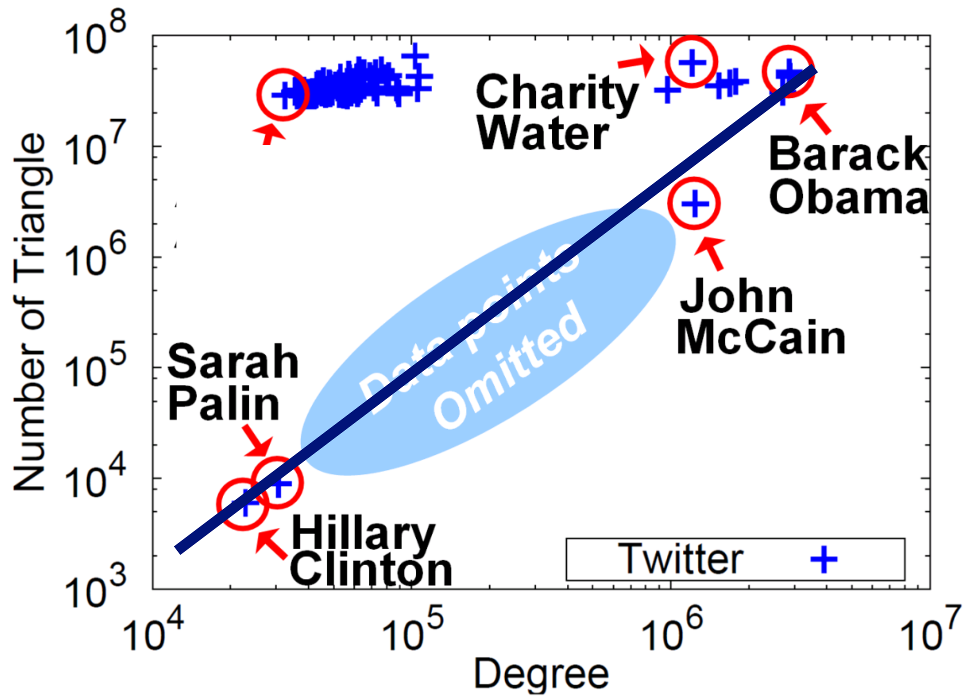
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

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

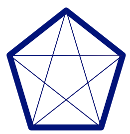
Triangle counting for large graphs?

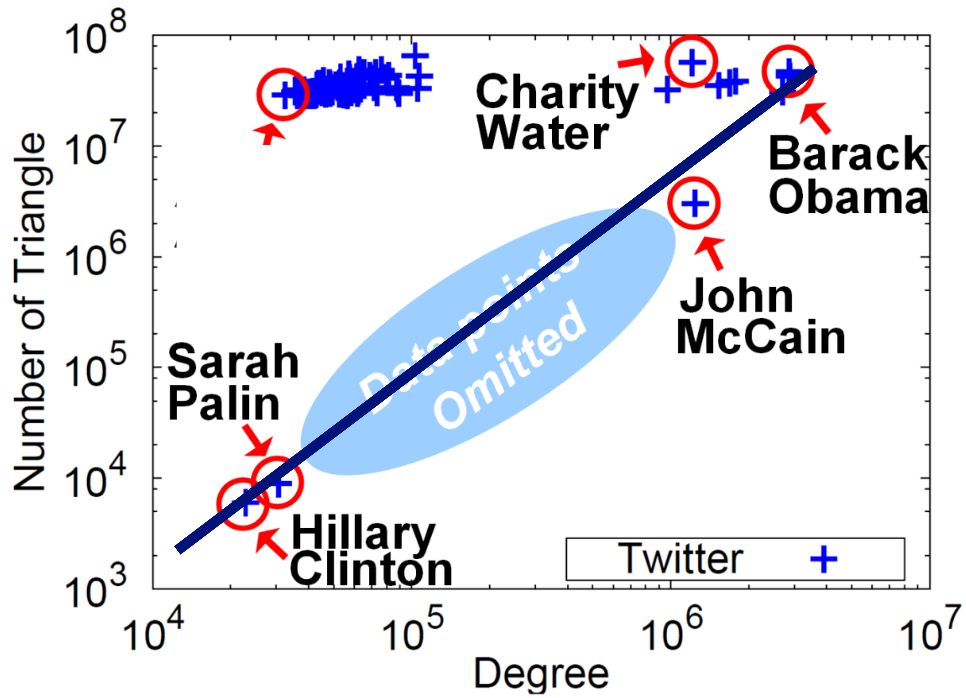


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?



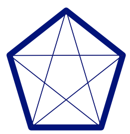


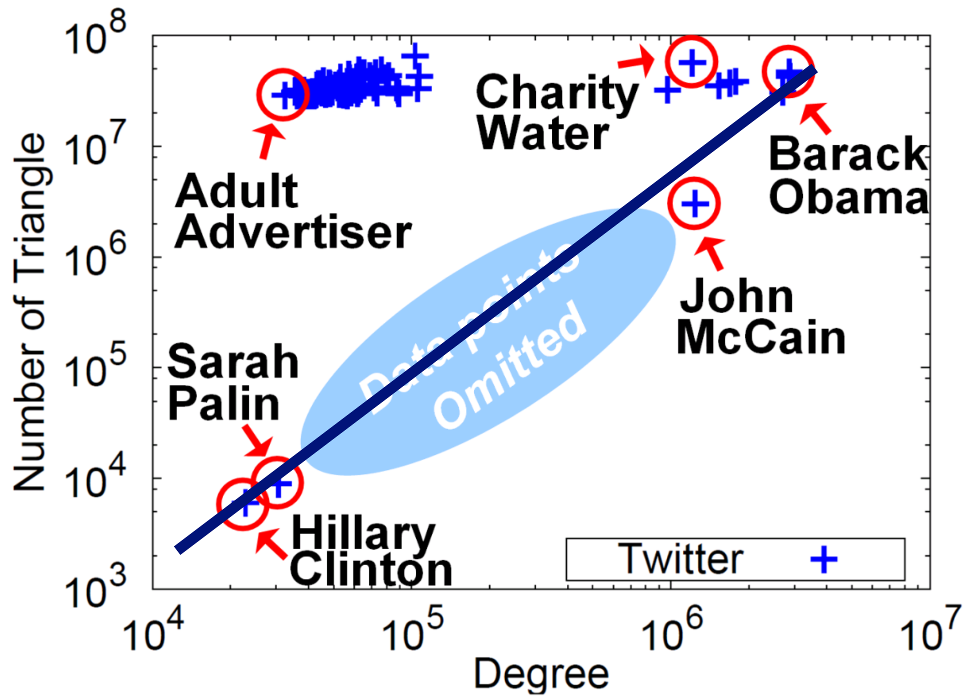
Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

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]

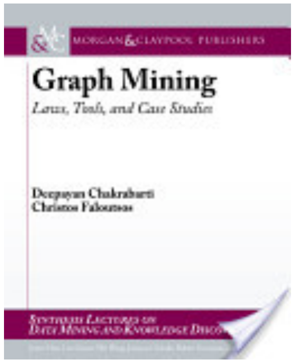
RTG: A Recursive Realistic Graph Generator using Random Typing Leman Akoglu and Christos Faloutsos. *PKDD*'09.

Carnegie Mellon

MORE Graph Patterns

	Unweighted	Weighted
Static	**L01.** Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04] **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]

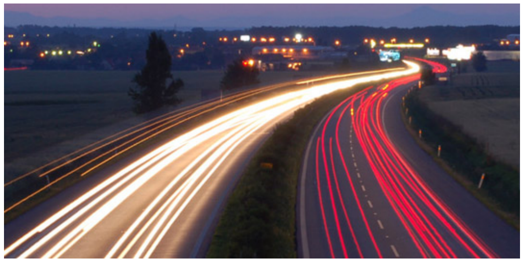
- Mary McGlohon, Leman Akoglu, Christos Faloutsos. *Statistical Properties of Social Networks.* in "Social Network Data Analytics" (Ed.: Charu Aggarwal)



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

http://www.cs.cmu.edu/~christos/TALKS/16-06-19-ICML/

Roadmap



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

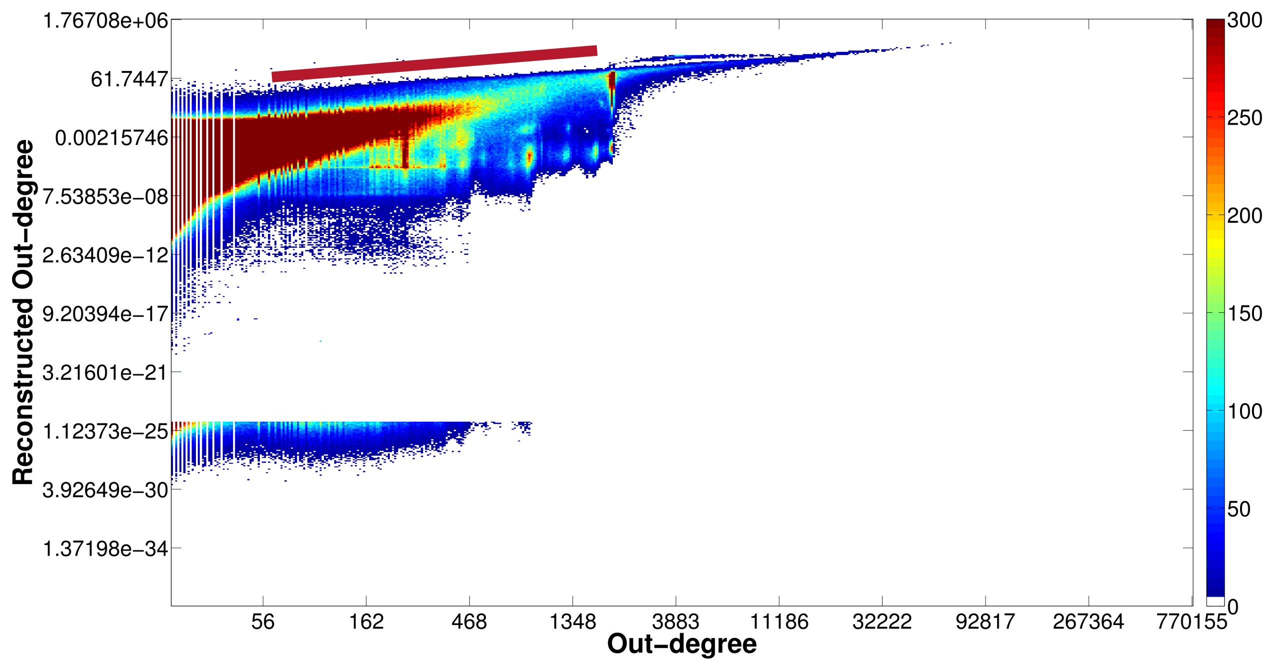
Problem: Social Network Link Fraud

Target: find "stealthy" attackers missed by other algorithms

Clique

41.7M nodes
1.5B edges

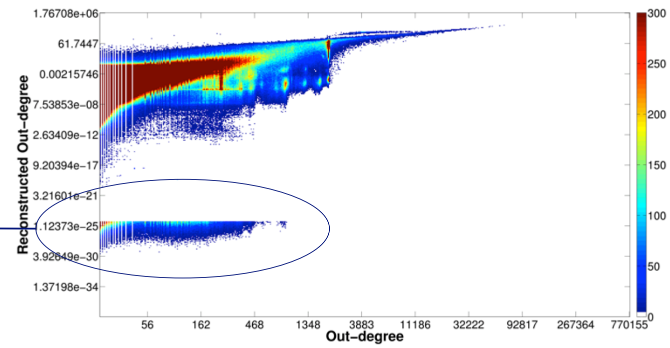
Bipartite core



Carnegie Mellon

Problem: Social Network Link Fraud

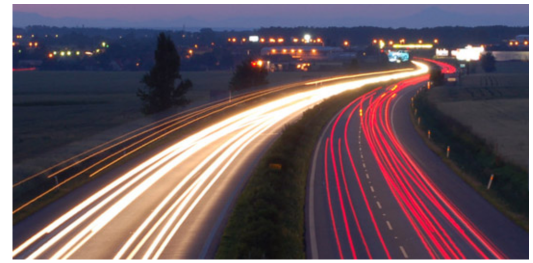
Target: find "stealthy" attackers missed by other algorithms



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



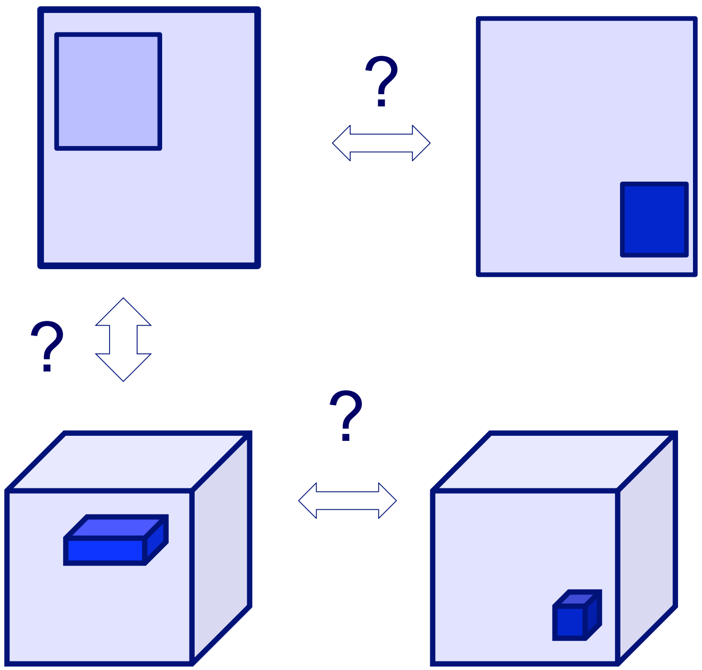
- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns
  - Anomaly / fraud detection
    - CopyCatch
    - Spectral methods (‘fBox’, **suspiciousness**)
    - Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

Carnegie Mellon

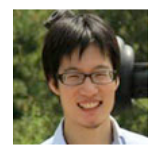
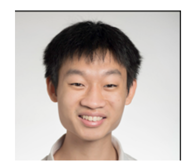
Suspicious Patterns in Event Data

2-modes

n-modes





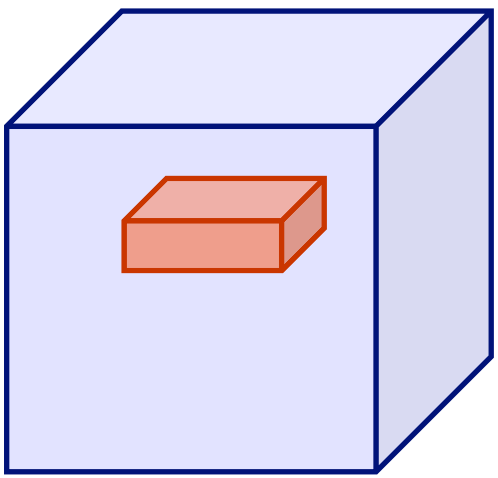



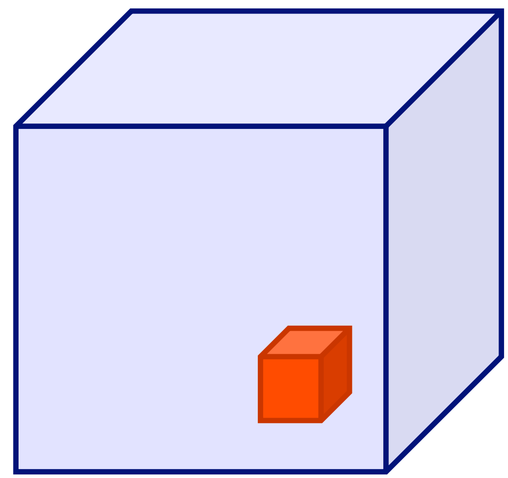
A General Suspiciousness Metric for Dense Blocks in Multimodal Data, Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos, *ICDM*, 2015.

Suspicious Patterns in Event Data

Which is more suspicious?

20,000 Users		225 Users
Retweeting same 20 tweets		Retweeting same 1 tweet
6 times each	vs.	15 times each
All in 10 hours		All in 3 hours
		All from 2 IP addresses

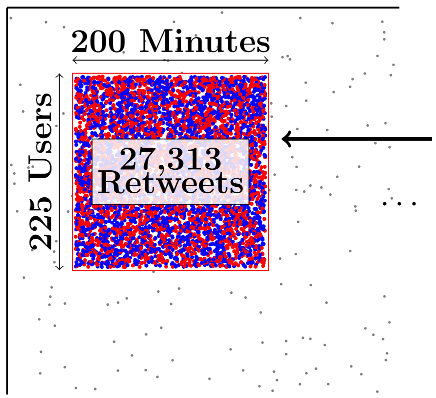


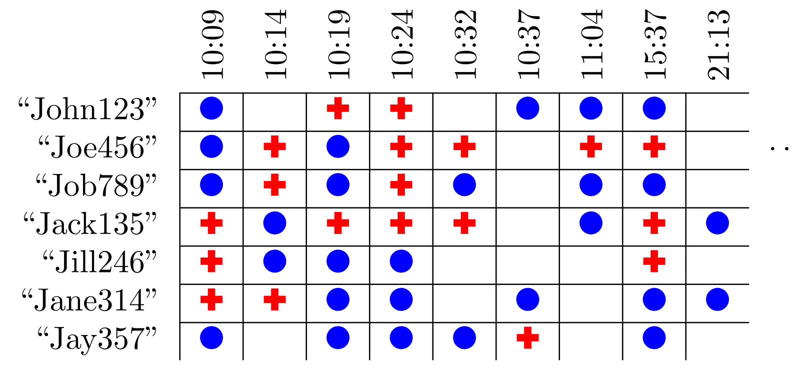


Google

Answer: volume * $D_{KL}(p || p_{background})$

Suspicious Patterns in Event Data

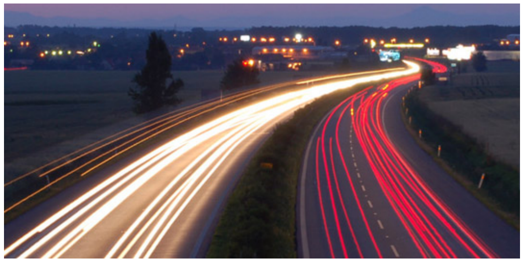




Retweeting: "Galaxy Note Dream Project: Happy Happy Life Traveling the World"

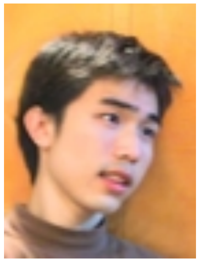
	#	User × tweet × IP × minute	Mass c	Suspiciousness
CROSSSPOT	1	14×1×2×1,114	41,396	1,239,865
	2	225×1×2×200	27,313	777,781
	3	8×2×4×1,872	17,701	491,323
HOSVD	1	24×6×11×439	3,582	131,113
	2	18×4×5×223	1,942	74,087
	3	14×2×1×265	9,061	381,211

Roadmap

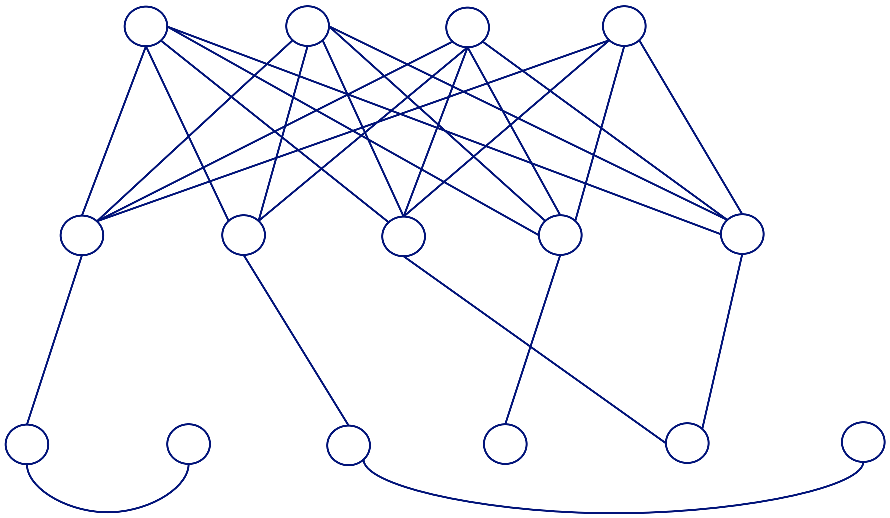


- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns
  - Anomaly / fraud detection
    - Spectral methods (‘fBox’)
    - High-density sub-matrices
    - Belief propagation
- Part#2: time-evolving graphs; tensors
- Part#3: time sequences
- Conclusions

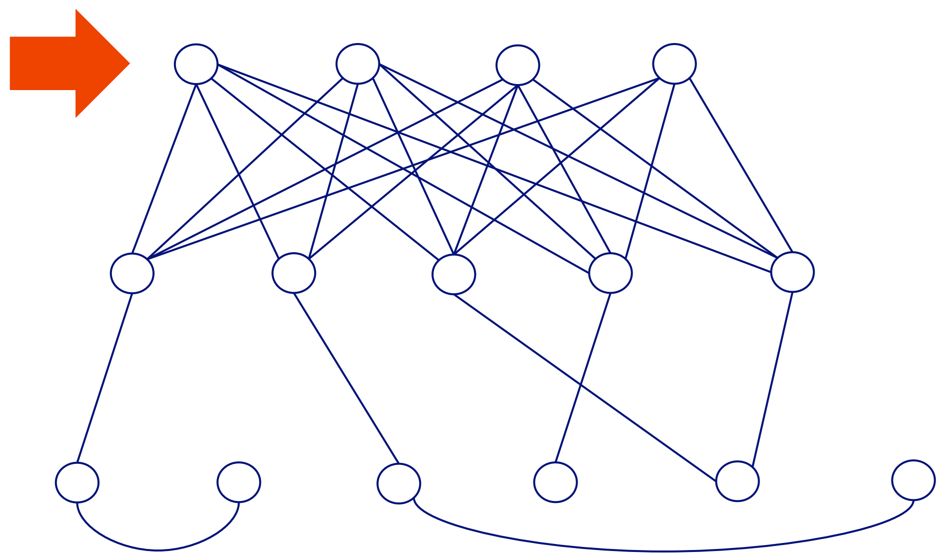
E-bay Fraud detection



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

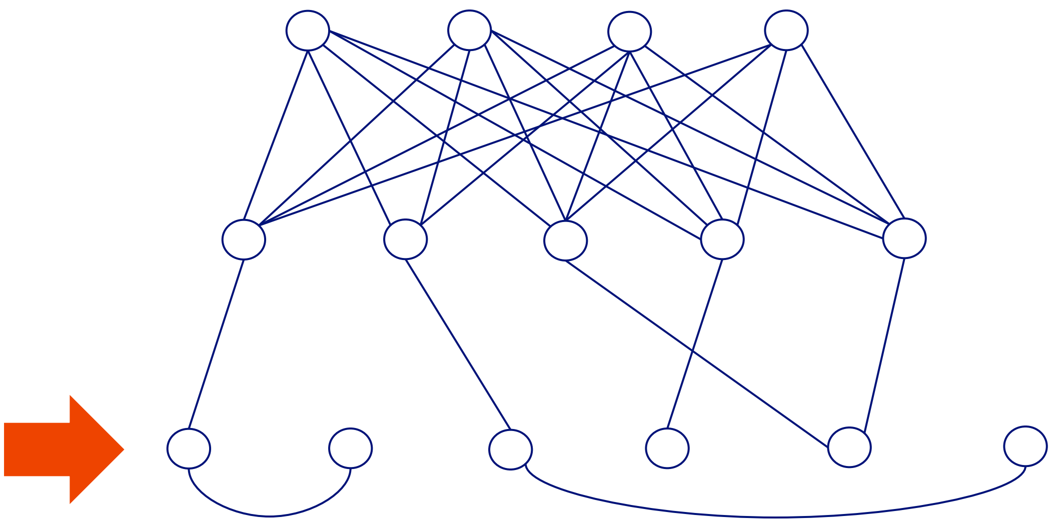


E-bay Fraud detection



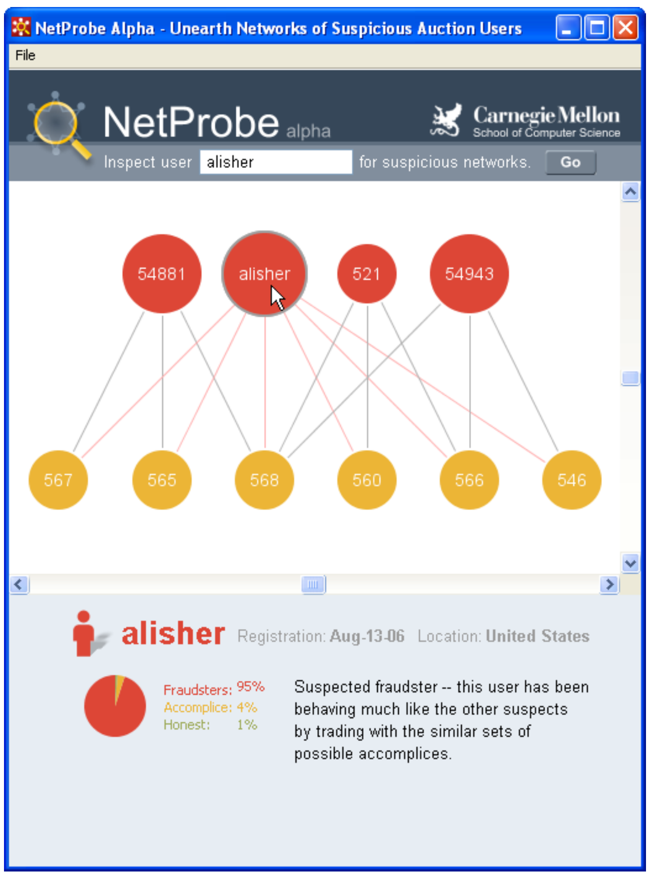
Carnegie Mellon

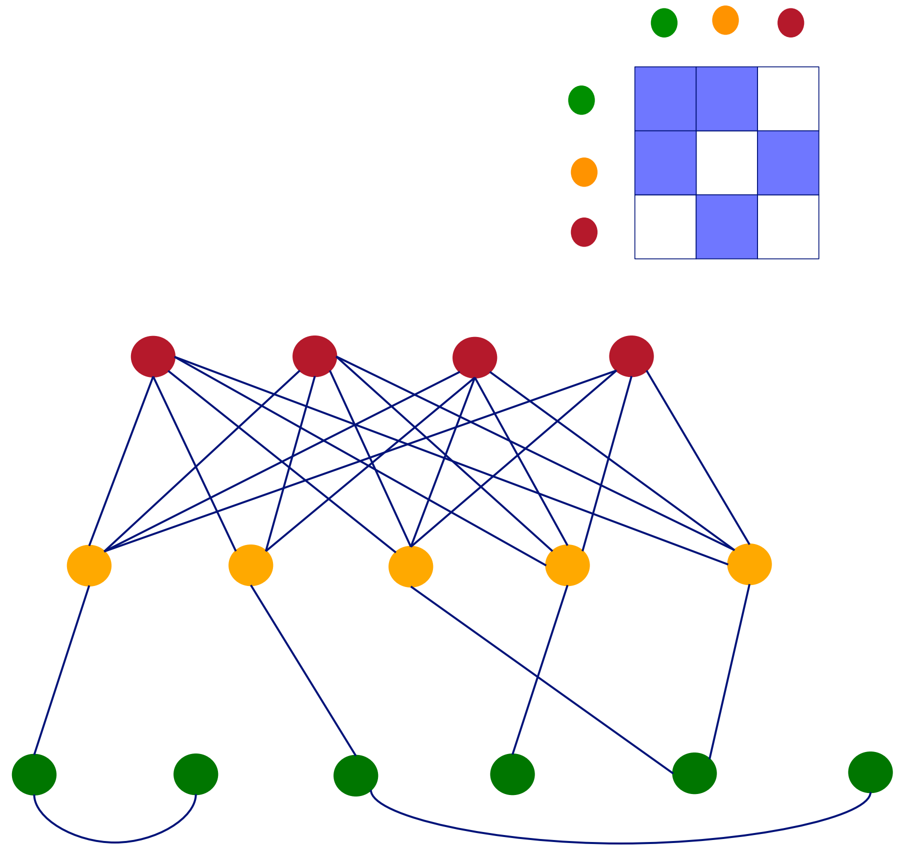
E-bay Fraud detection



Carnegie Mellon

E-bay Fraud detection - NetProbe





Popular press

USA TODAY

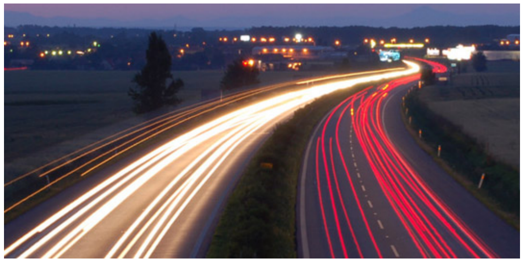
The Washington Post

Los Angeles Times

And less desirable attention:

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

Roadmap



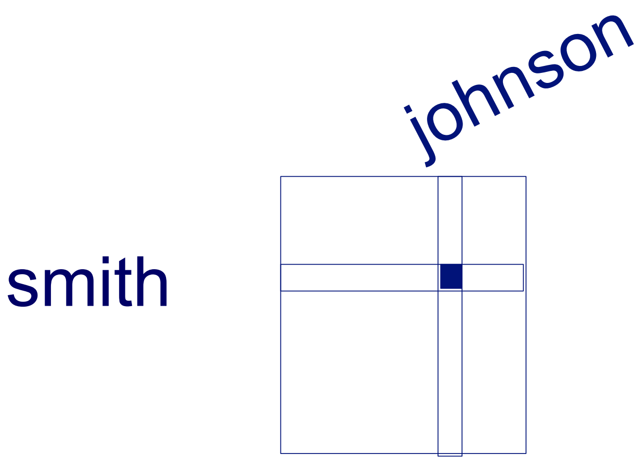
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
  - P2.1: time-evolving graphs
  - [P2.2: with side information ('coupled' M.T.F.)
  - Speed]
- Part#3: time sequences
- Conclusions

Carnegie Mellon

Part 2: Time evolving graphs; tensors

Graphs over time -> tensors!

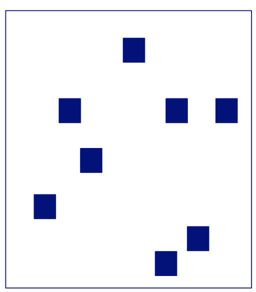
- Problem #2.1:
  - Given who calls whom, and when
  - Find patterns / anomalies



Carnegie Mellon

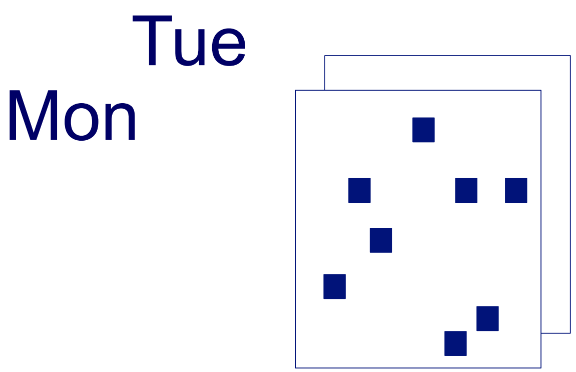
Graphs over time -> tensors!

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



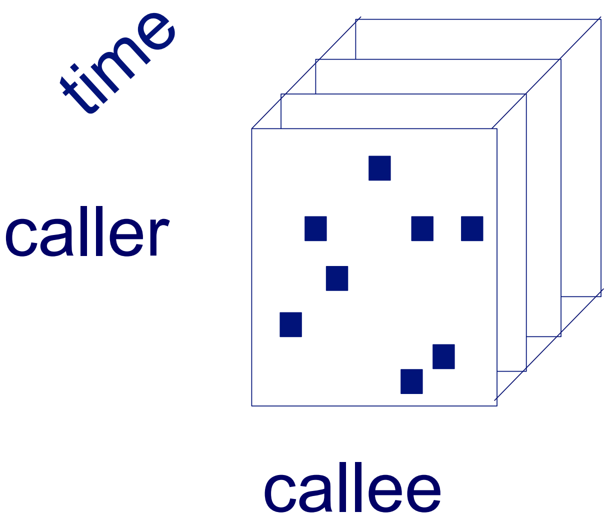
Graphs over time -> tensors!

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



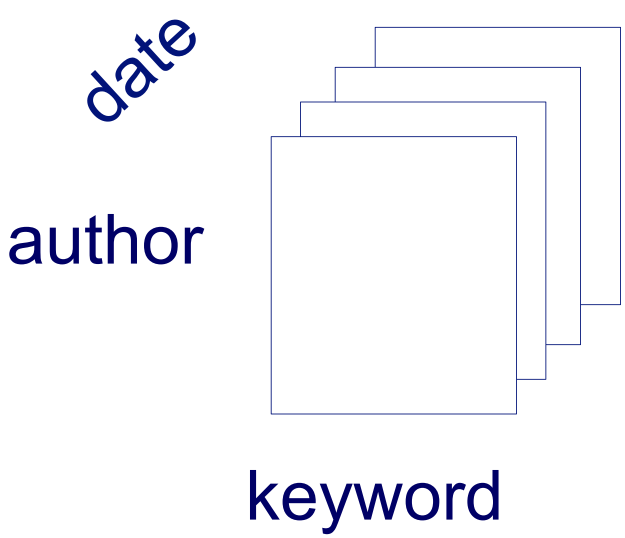
Graphs over time -> tensors!

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



Graphs over time -> tensors!

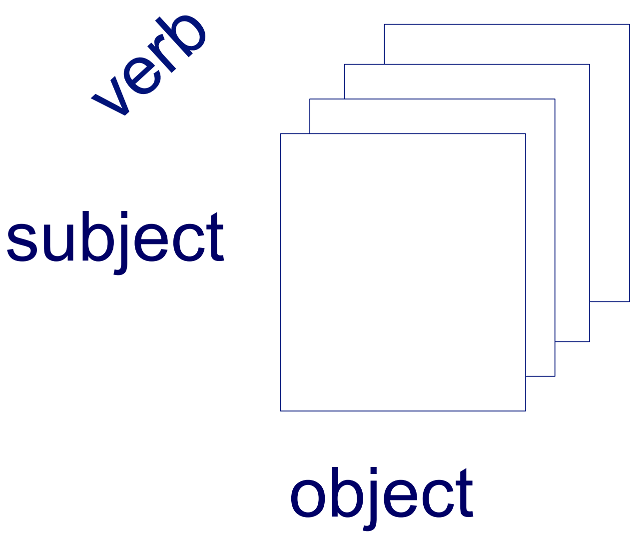
- Problem #2.1’:
  - Given author-keyword-date
  - Find patterns / anomalies



MANY more settings, with >2 ‘modes’

Graphs over time -> tensors!

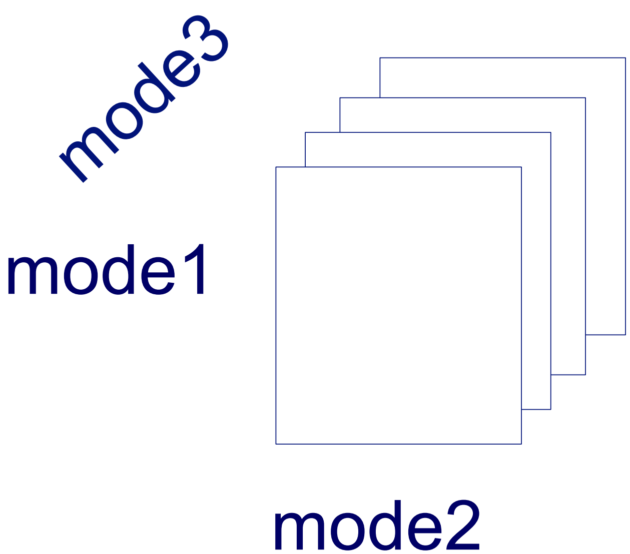
- Problem #2.1'':
  - Given subject – verb – object facts
  - Find patterns / anomalies



MANY more settings, with >2 'modes'

Graphs over time -> tensors!

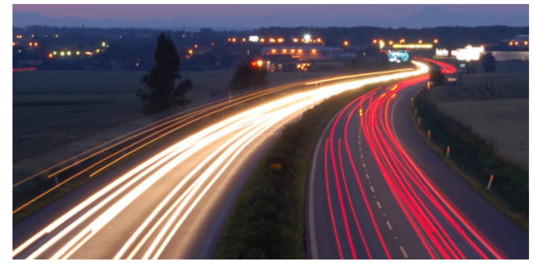
- Problem #2.1''':
  - Given <triplets>
  - Find patterns / anomalies



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

Carnegie Mellon

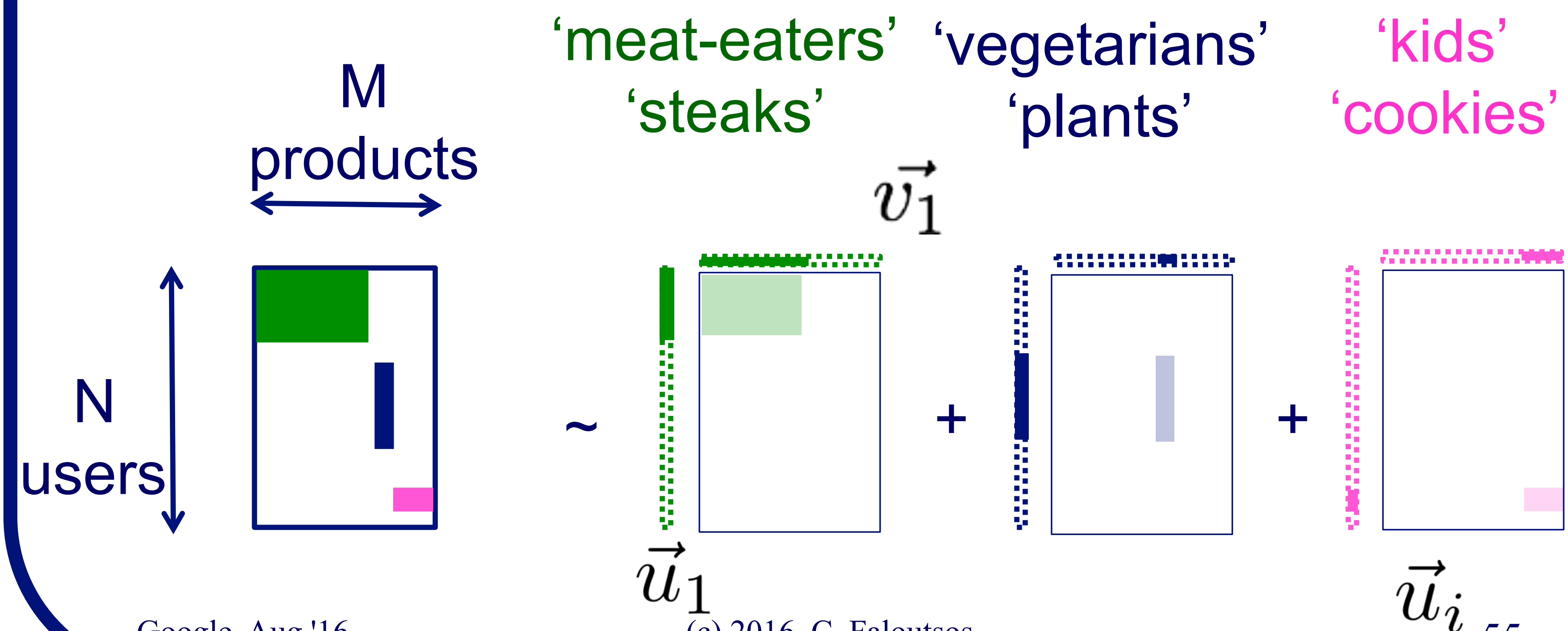
Roadmap



- 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]
- Conclusions

Answer to both: tensor factorization

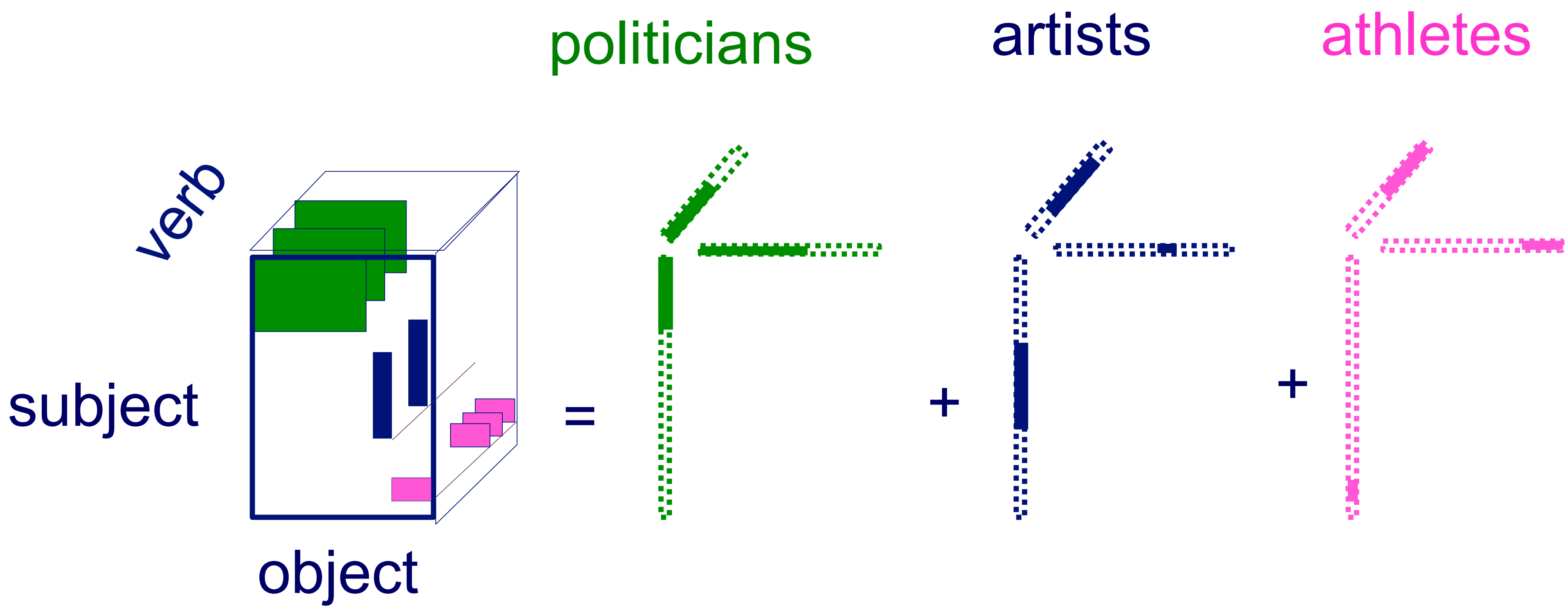
- Recall: (SVD) matrix factorization: finds blocks



Carnegie Mellon

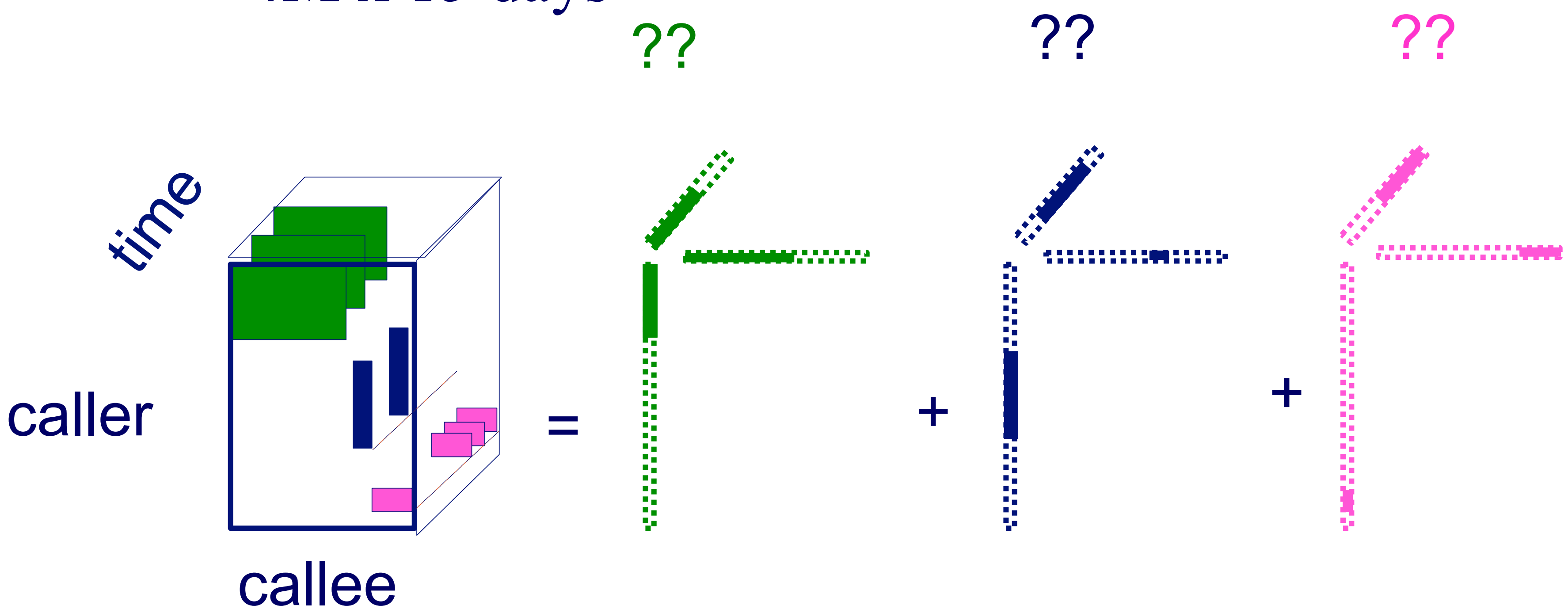
Answer to both: tensor factorization

- PARAFAC decomposition

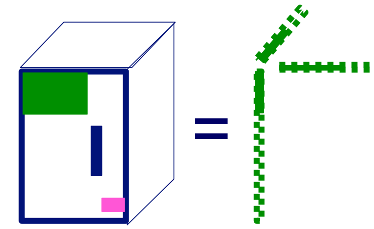


Answer: tensor factorization

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

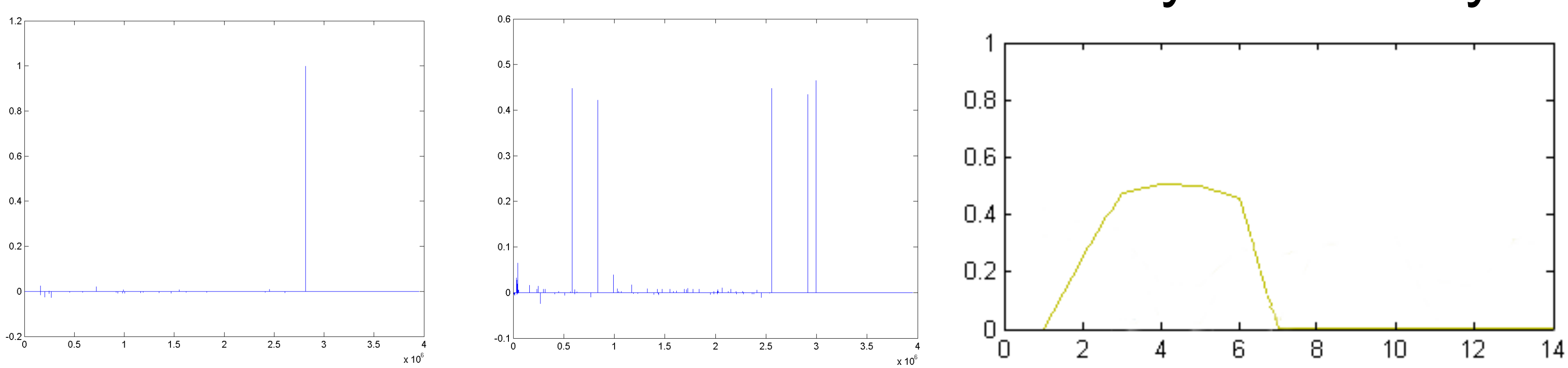


Anomaly detection in time-evolving graphs



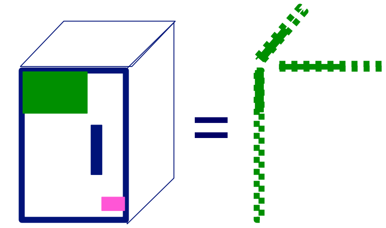
- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

1 caller | 5 receivers | 4 days of activity



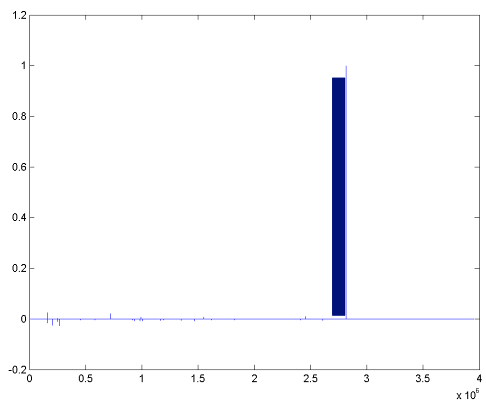
~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs

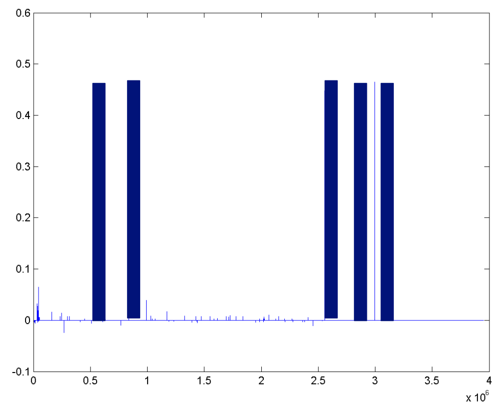


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

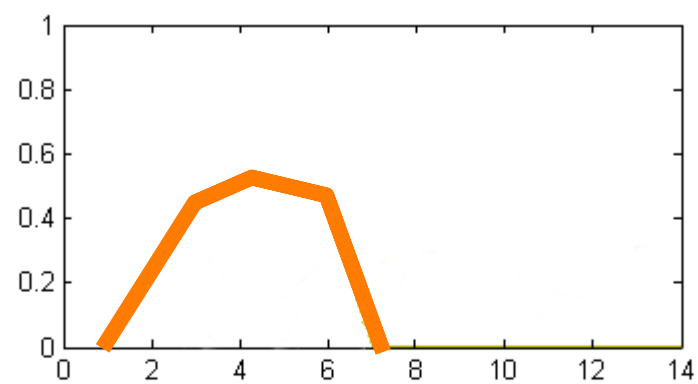
1 caller



5 receivers



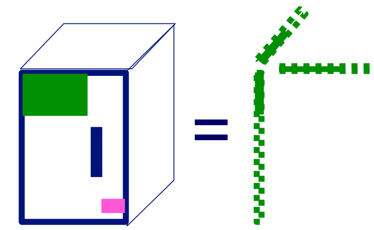
4 days of activity



~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs



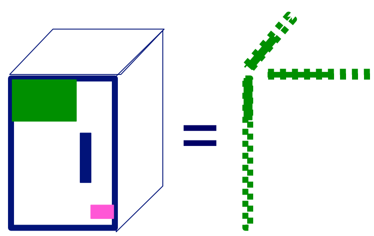
- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

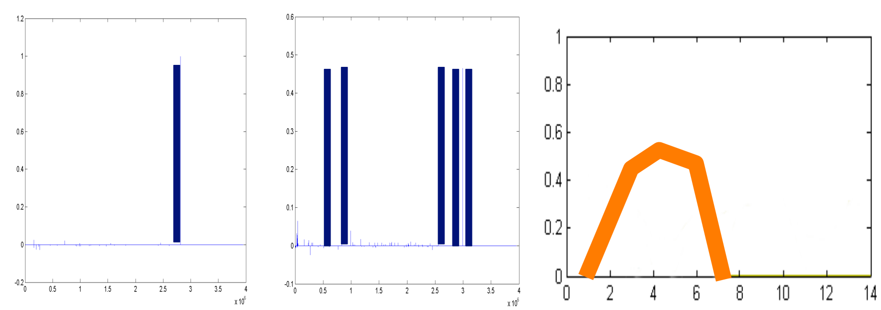
Miguel Araujo, Spiros Papadimitriou, Stephan Günnemann, Christos Faloutsos, Prithwish Basu, Ananthram Swami, Evangelos Papalexakis, Danai Koutra. *Com2: Fast Automatic Discovery of Temporal (Comet) Communities*. PAKDD 2014, Tainan, Taiwan.

Carnegie Mellon

Part 2: Conclusions

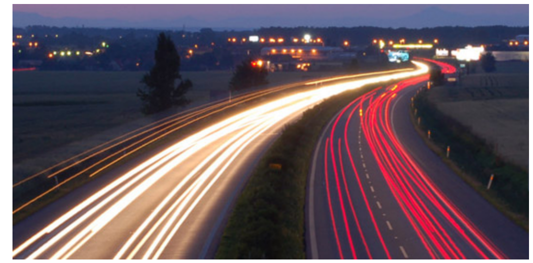
- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- (GigaTensor/HaTen2 -> fast & scalable)





Part 3: Time sequences

Roadmap



- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Part#3: time sequences
- Acknowledgements and Conclusions

Carnegie Mellon

Universidade de São Paulo

筑波大学 University of Tsukuba

Carnegie Mellon University

KDD 2015 – Sydney, Australia

RSC: Mining and Modeling Temporal Activity in Social Media

Alceu F. Costa* Yuto Yamaguchi Agma J. M. Traina

Caetano Traina Jr. Christos Faloutsos

*alceufc@icmc.usp.br

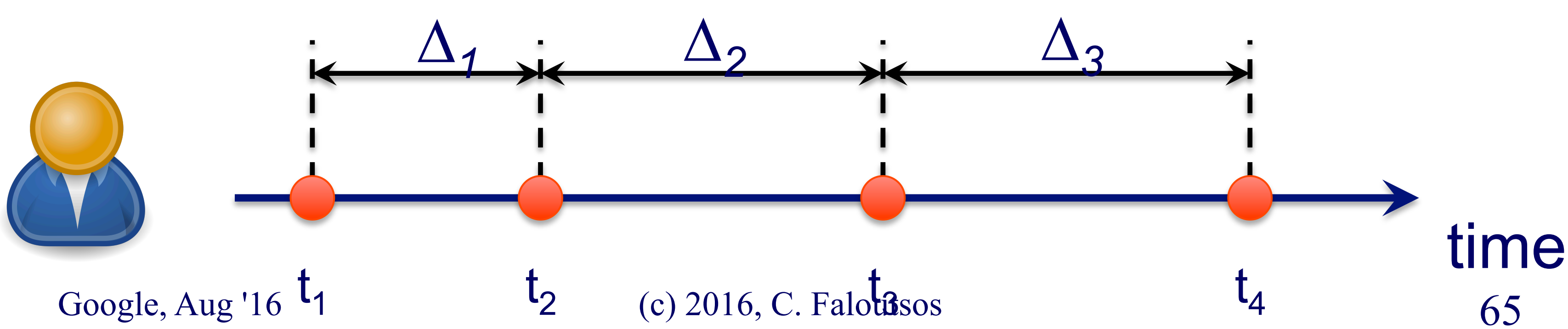
Pattern Mining: Datasets

Reddit Dataset	Twitter Dataset
Time-stamp from comments	Time-stamp from tweets
21,198 users	6,790 users
20 Million time-stamps	16 Million time-stamps

For each user we have:

Sequence of postings time-stamps: $\boldsymbol{T} = (t_1, t_2, t_3, \ldots)$

Inter-arrival times (IAT) of postings: $(\Delta_1, \Delta_2, \Delta_3, \ldots)$

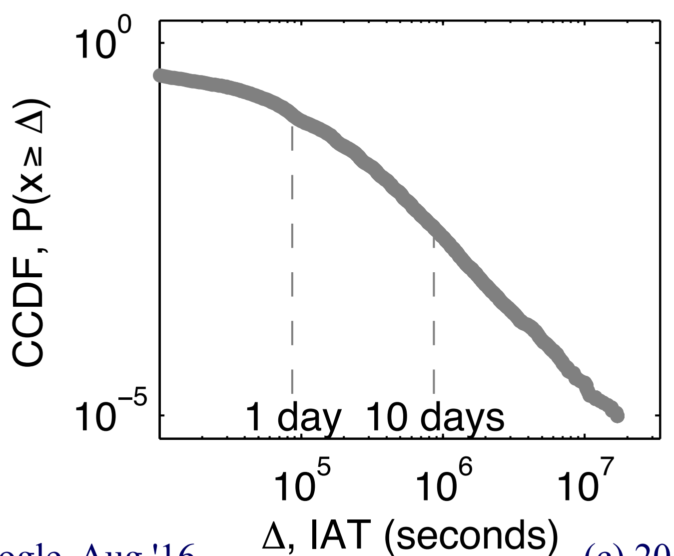


Pattern Mining

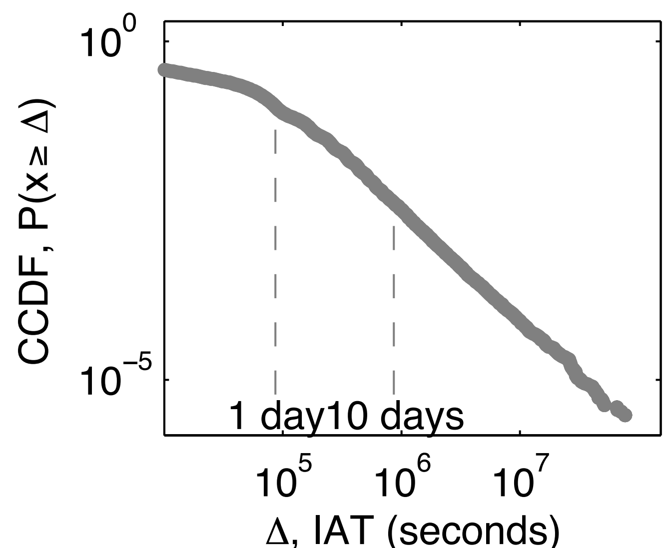
Pattern 1: Distribution of IAT is heavy-tailed

Users can be inactive for long periods of time before making new postings

IAT Complementary Cumulative Distribution Function (CCDF)
(log-log axis)



Reddit Users



Twitter Users

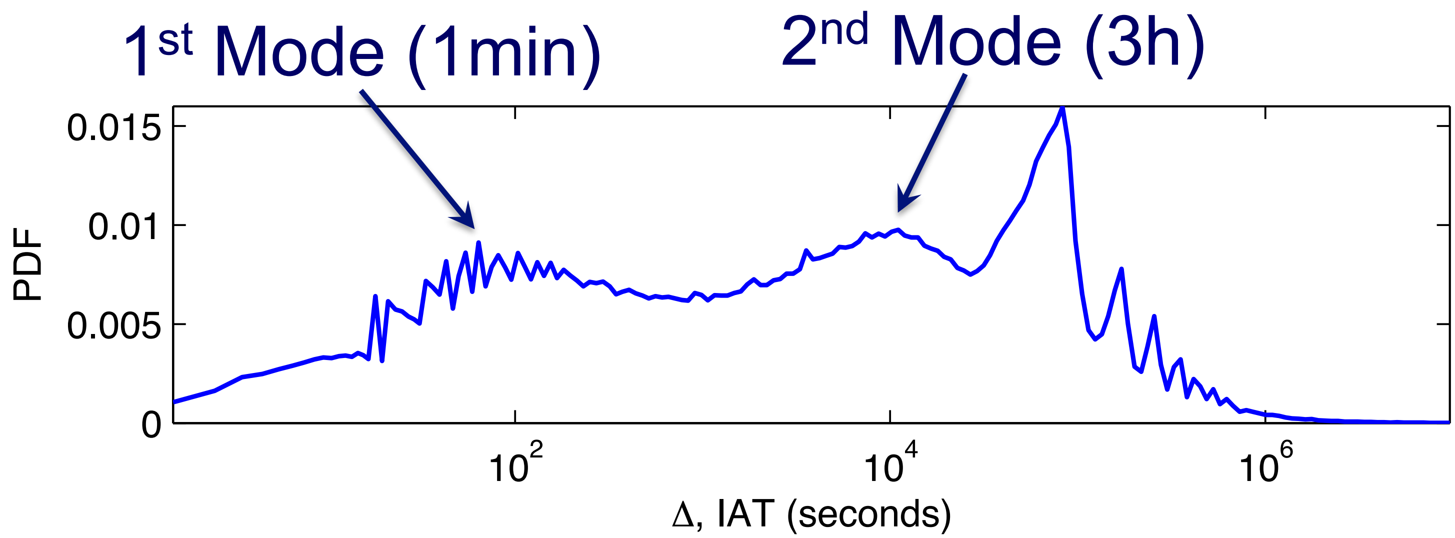
Carnegie Mellon

Pattern Mining

Pattern 2: Bimodal IAT distribution

'Active'/ 'resting' periods

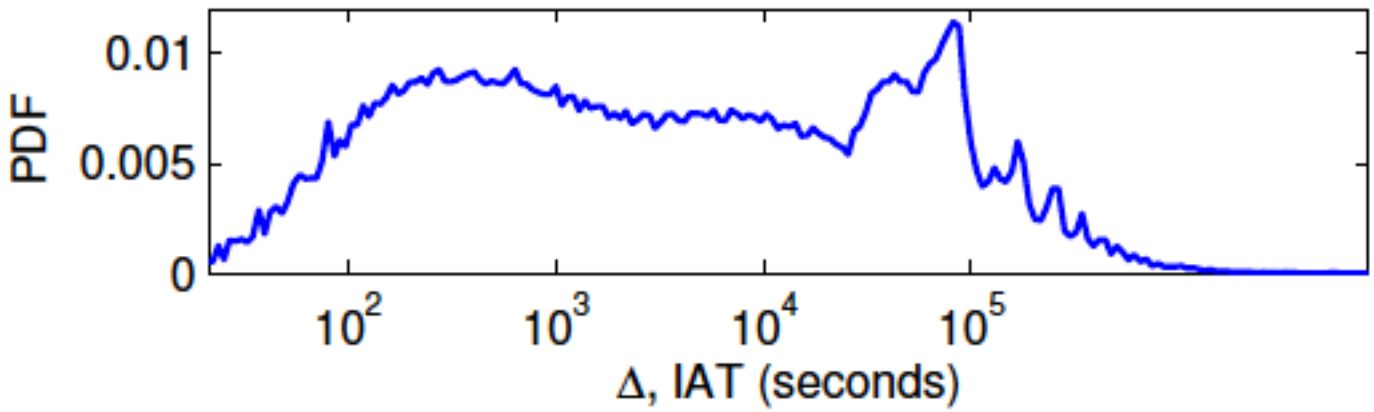
Log-binned histogram of postings IAT



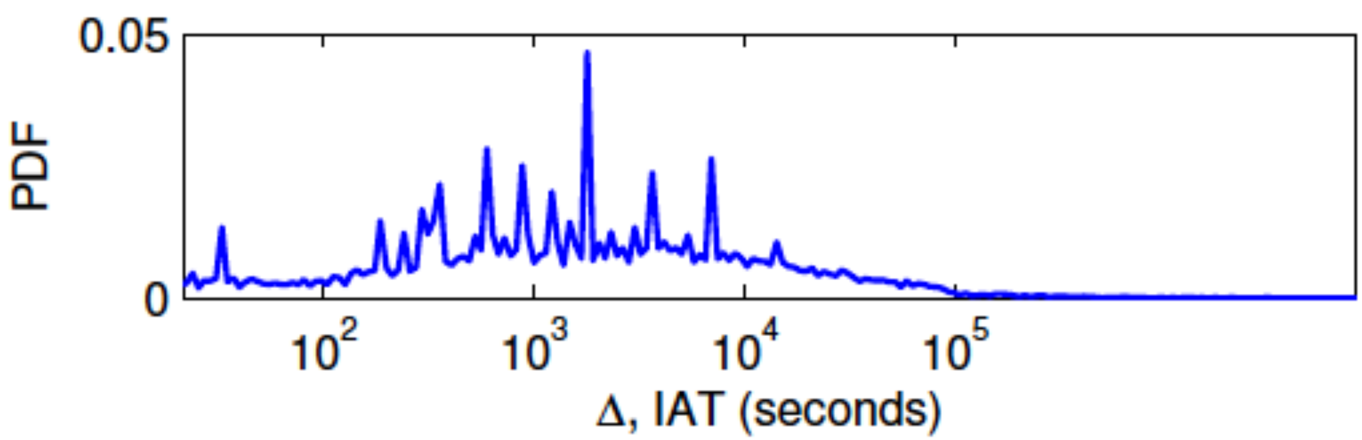
Twitter Users

Human? Robots?

linear

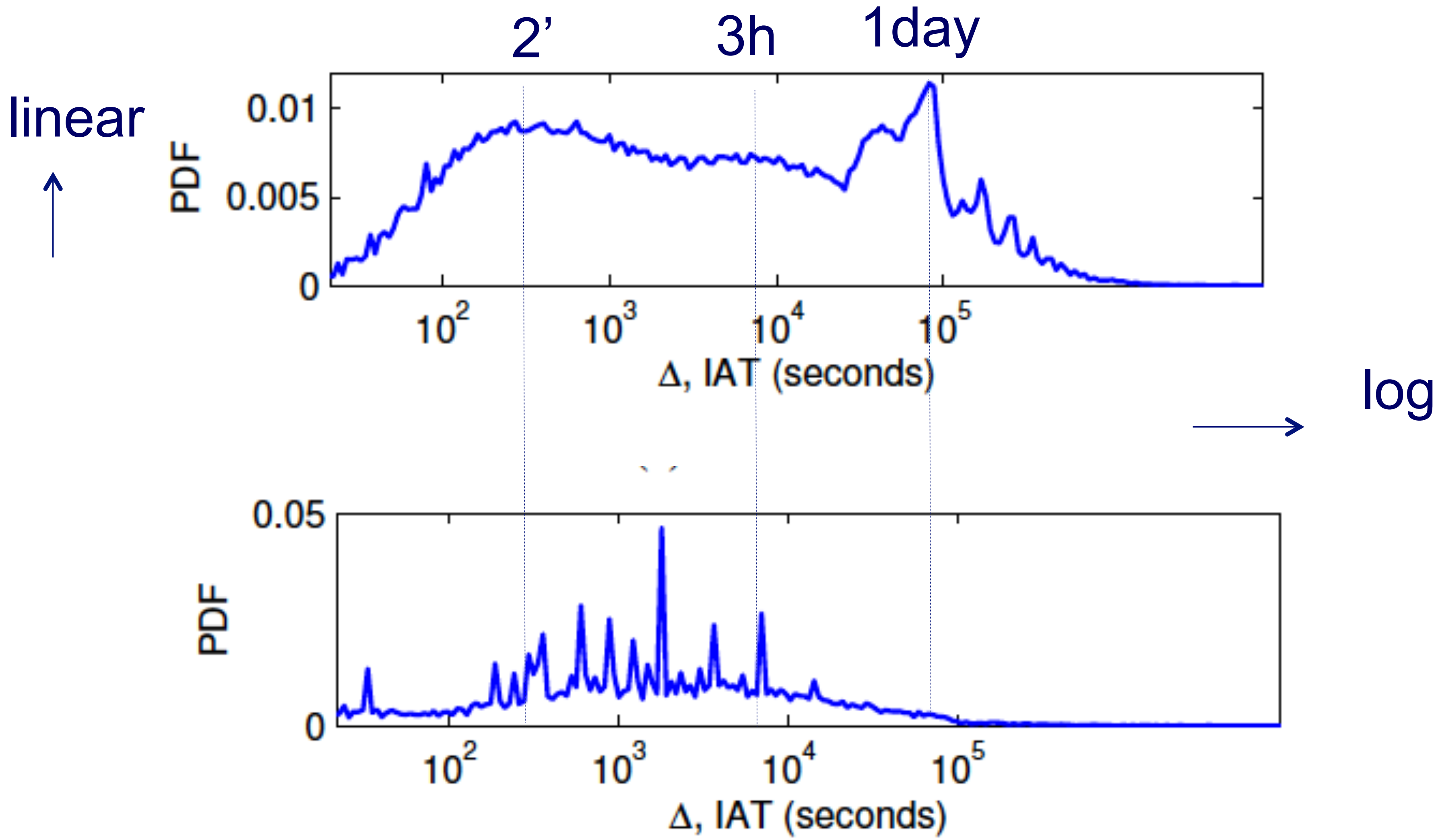


log



Carnegie Mellon

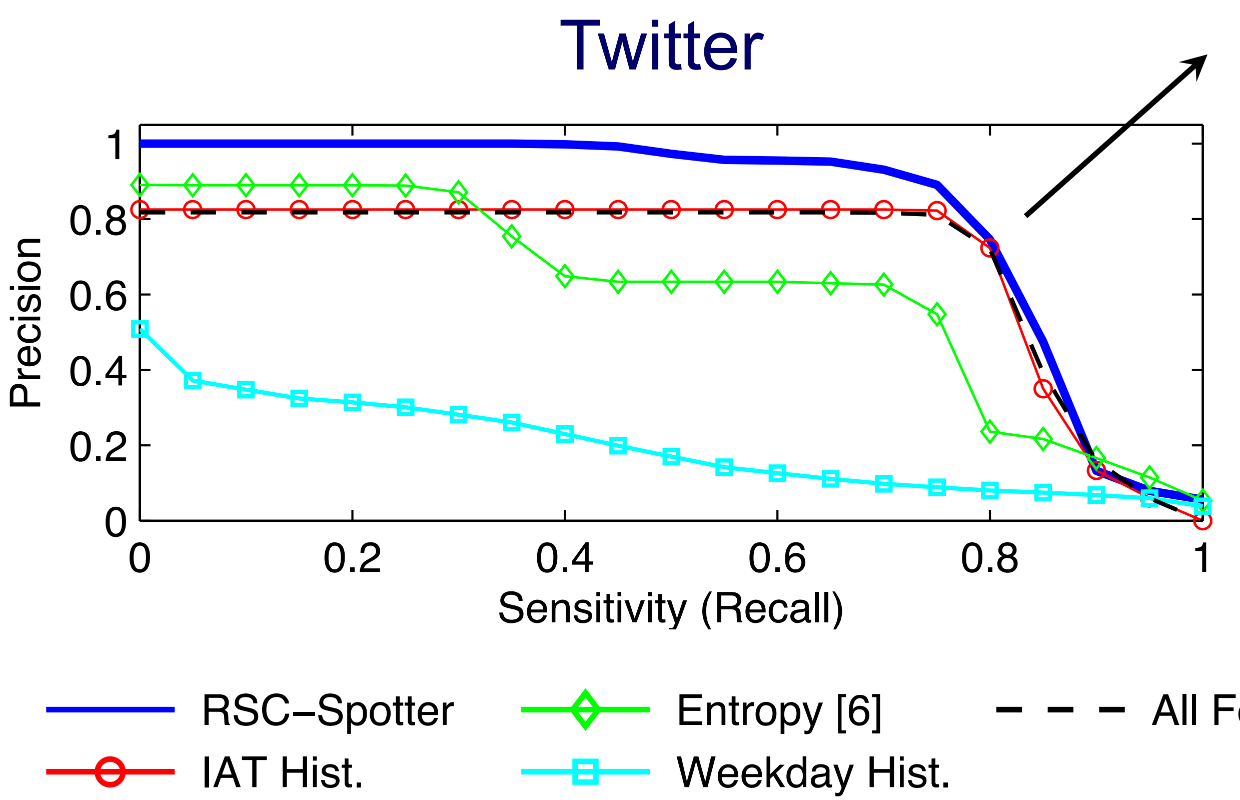
Human? Robots?



Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top



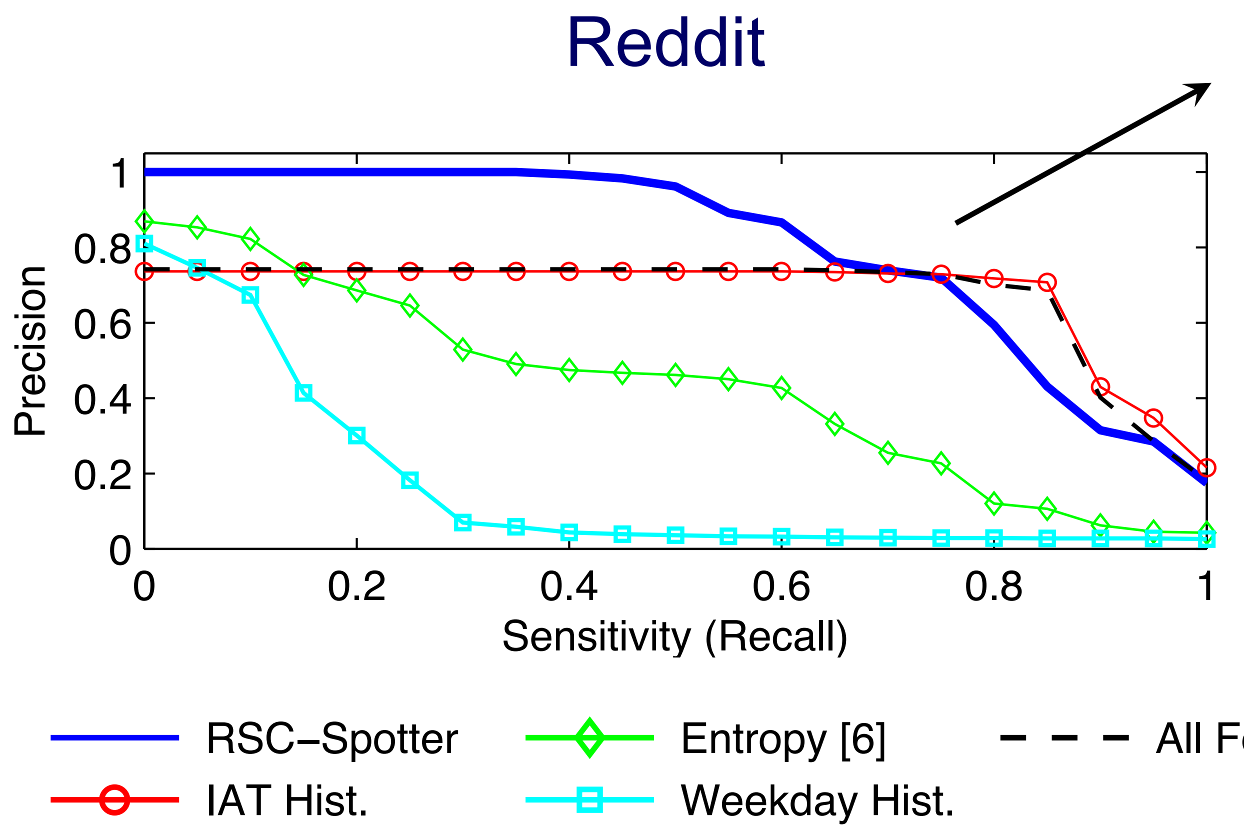
Precision > 94%
Sensitivity > 70%

With strongly imbalanced datasets
humans >> # bots

Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

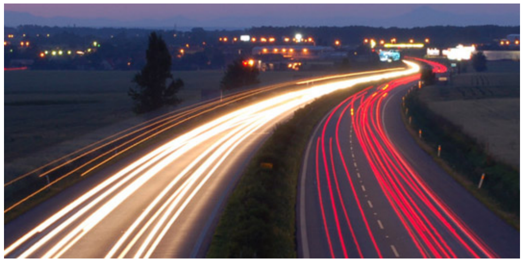
Good performance: curve close to the top



Precision > 96%
Sensitivity > 47%

With strongly imbalanced datasets
humans >> # bots

Roadmap



- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Part#3: time sequences
- 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

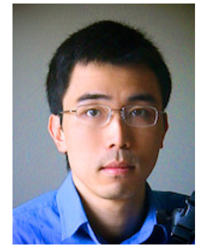
Carnegie Mellon

Cast

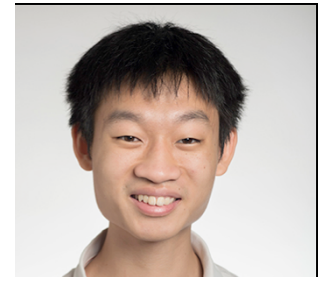
Akoglu, Leman

Araujo, Miguel

Beutel, Alex



Chau, Polo



Hooi, Bryan

Kang, U

Koutra, Danai

Papalexakis, Vagelis

Shah, Neil

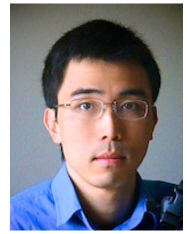
Song, Hyun Ah

Cast

Akoglu, Leman

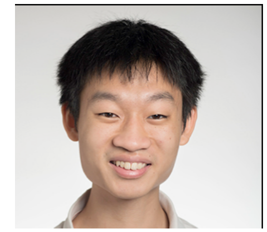
Araujo, Miguel

Beutel, Alex



Chau, Polo

Eswaran, Dhivya

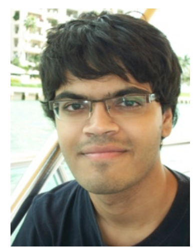


Hooi, Bryan

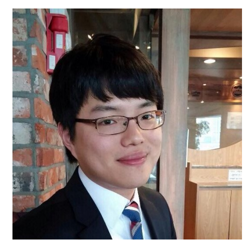
Kang, U

Koutra, Danai

Papalexakis, Vagelis



Shah, Neil

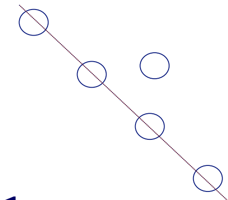


Shin, Kijung

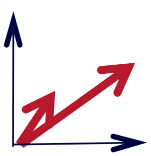
Song, Hyun Ah

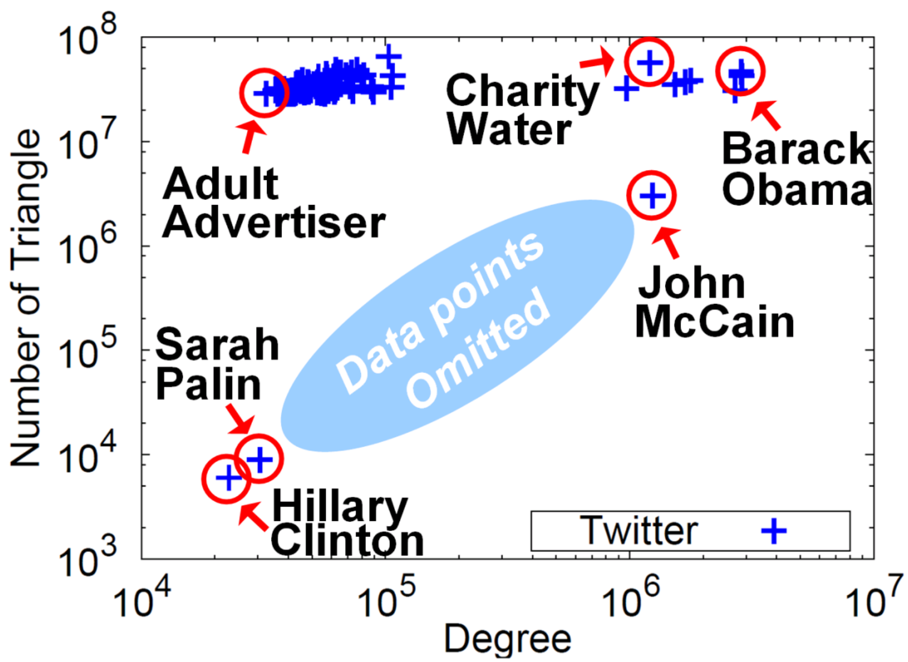
CONCLUSION#1 – Big data

- **Patterns**  **Anomalies**



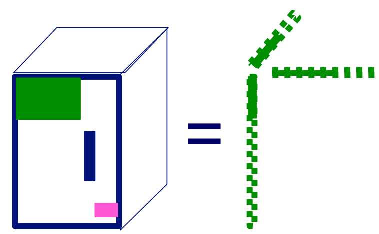
- **Large** datasets reveal patterns/outliers that are invisible otherwise

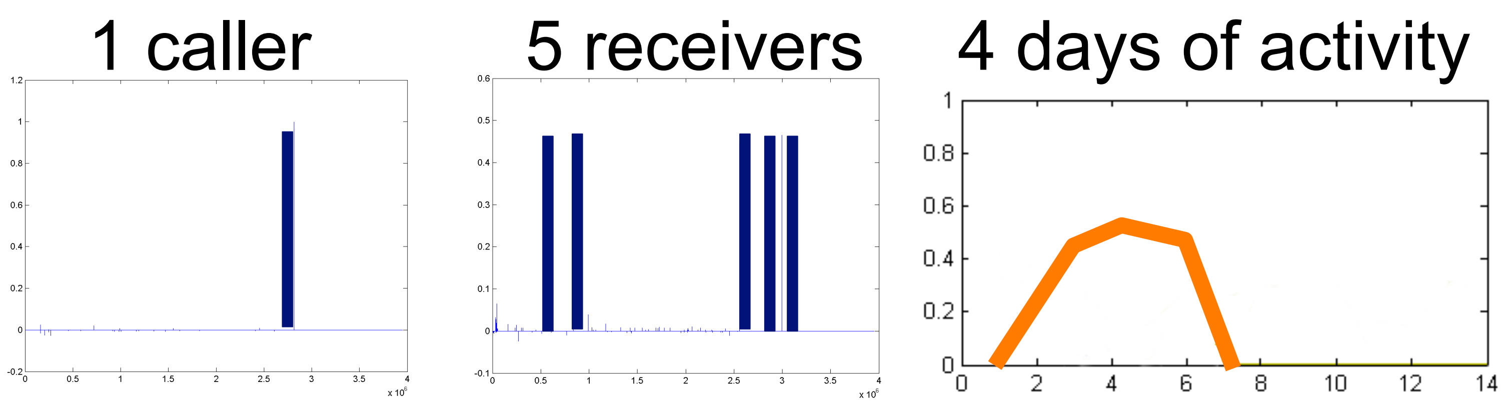




CONCLUSION#2 – tensors

- powerful tool

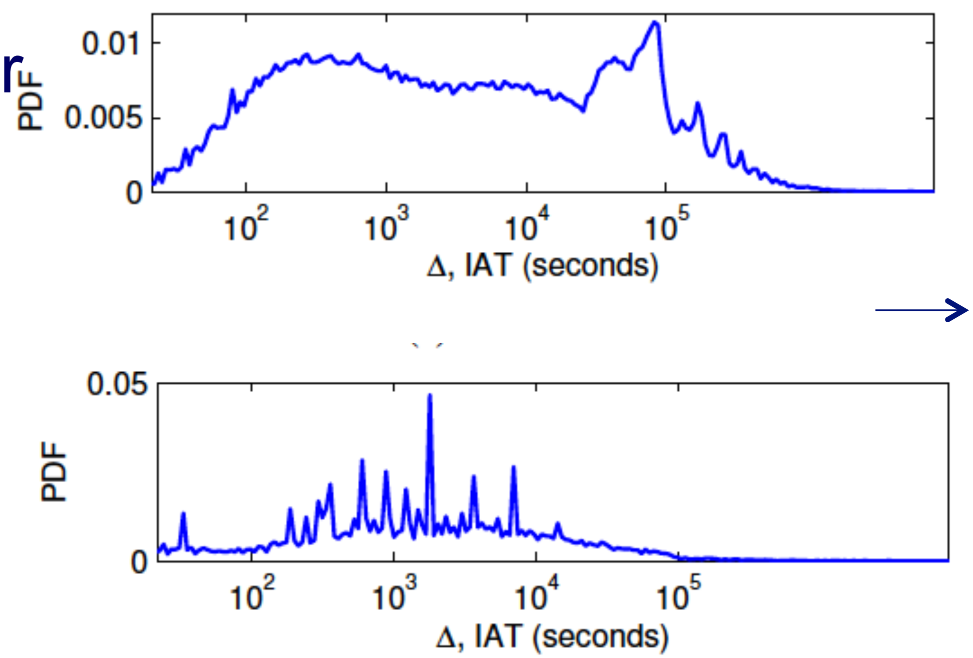




Conclusion#3

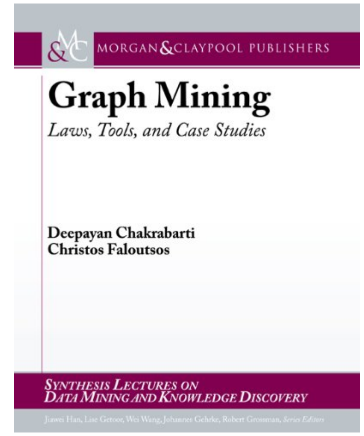
- Different footprints of real vs ‘robot’ users

linear



log

References

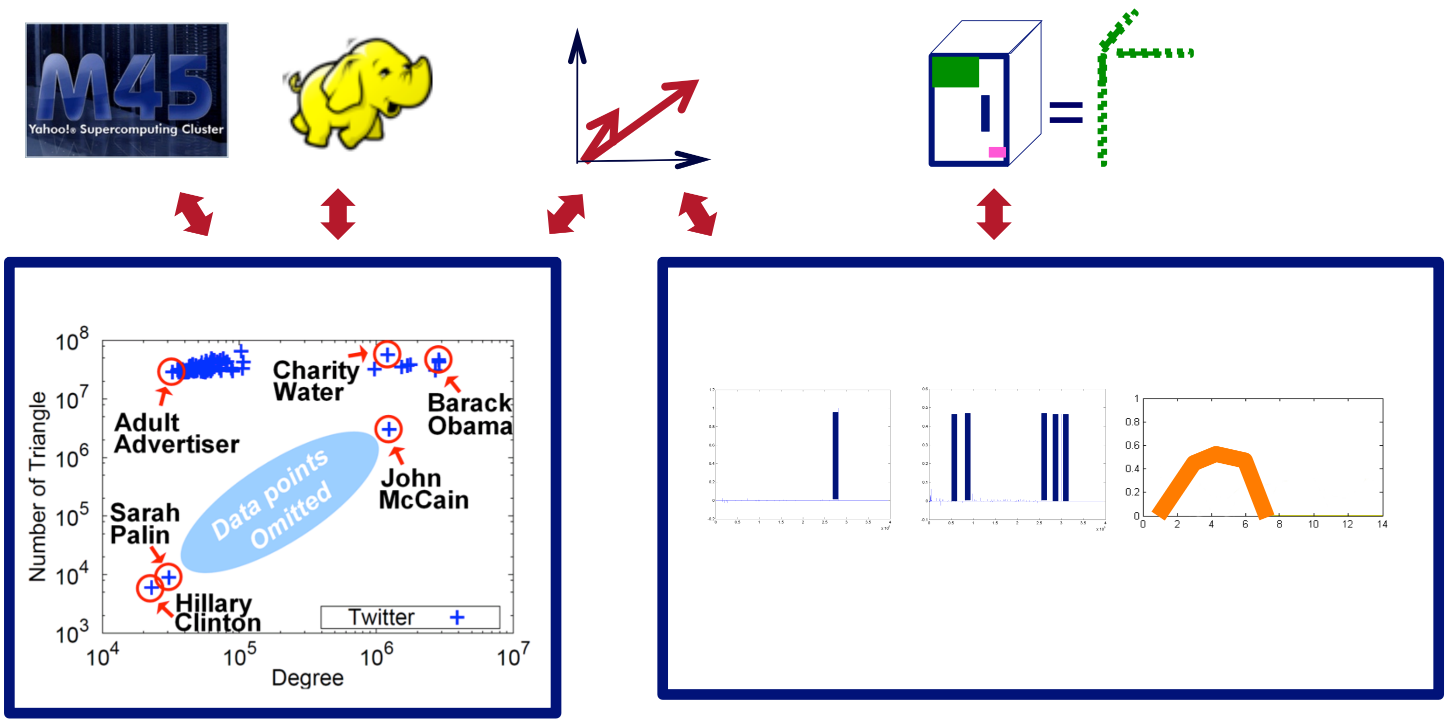


- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
- http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006

- *Graph-based Anomaly Detection and Description: A Survey*, Leman Akoglu, Hanghang Tong, Danai Koutra
- http://arxiv.org/abs/1404.4679

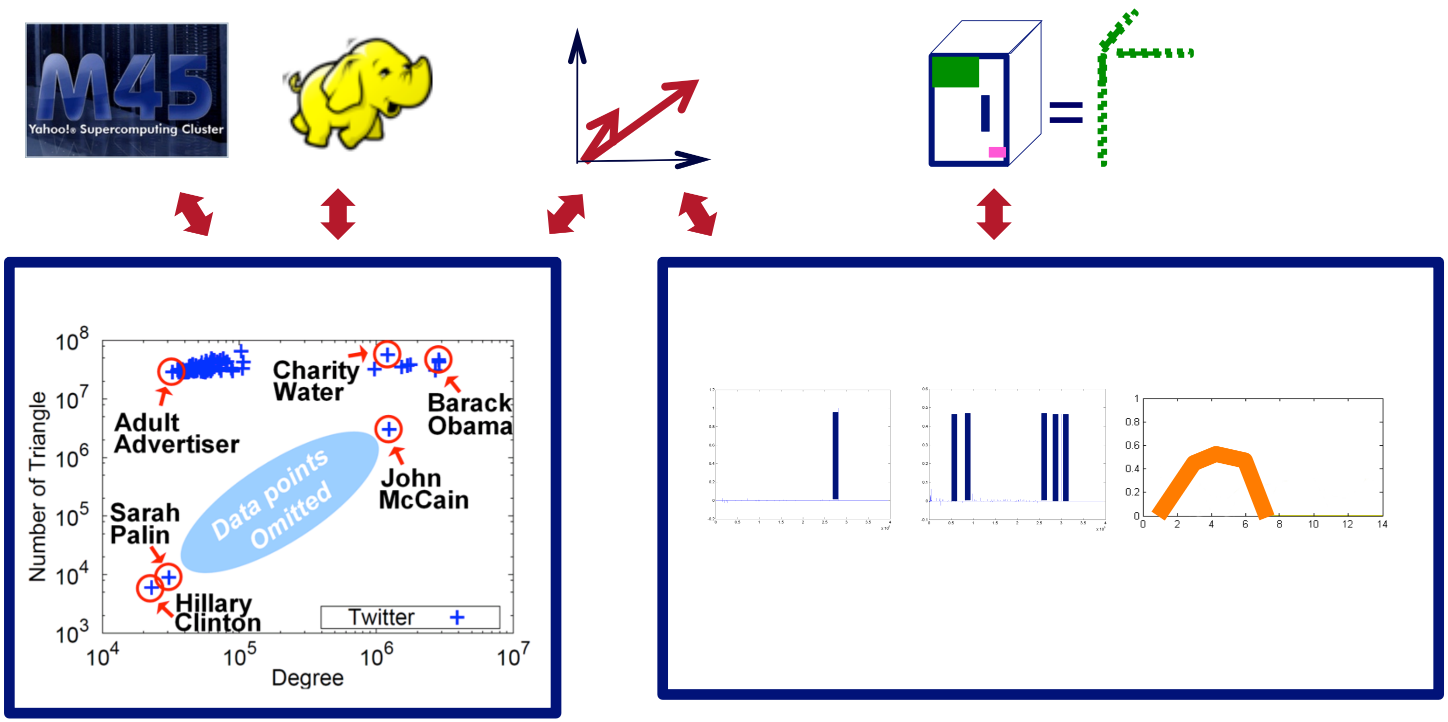
TAKE HOME MESSAGE:

Cross-disciplinarity



Thank you!

Cross-disciplinarity



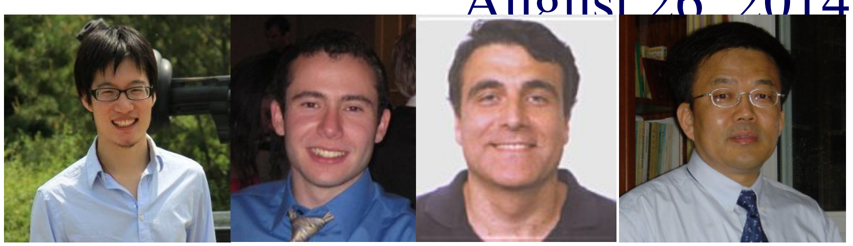
Carnegie Mellon

Catchsync: catch synchronized behavior in large directed graphs

Meng Jiang

Joint work with Peng Cui, Alex Beutel, Christos Faloutsos and Shiqiang Yang

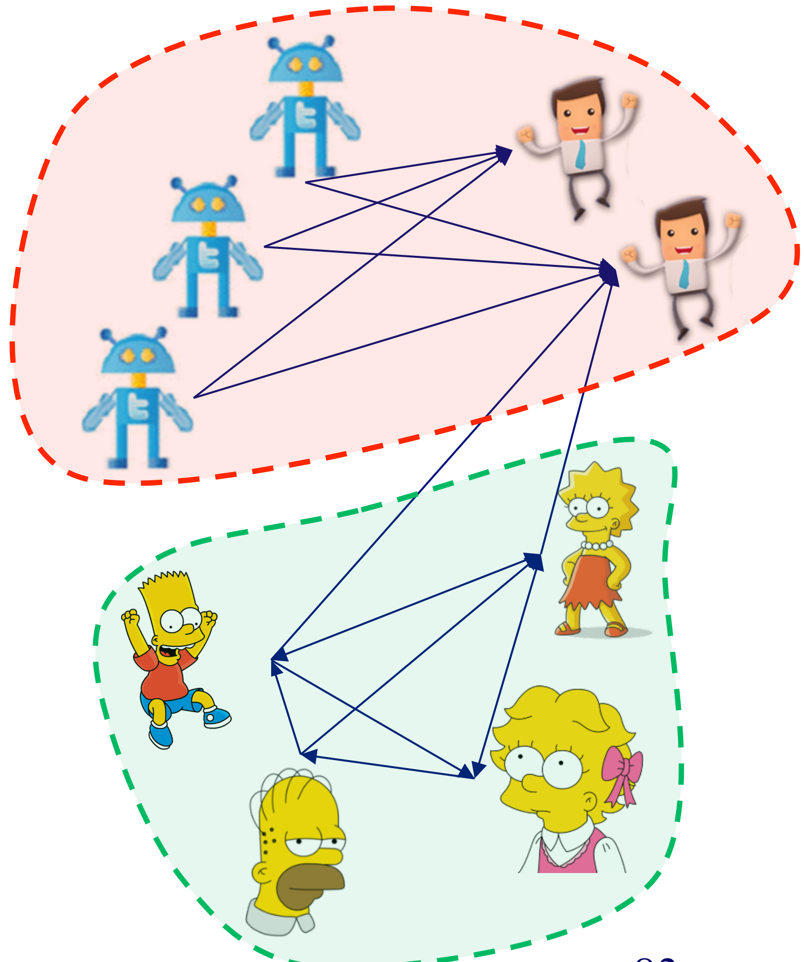
August 26, 2014 – NYC, USA



Fraud Detection: Graph Analysis Problem

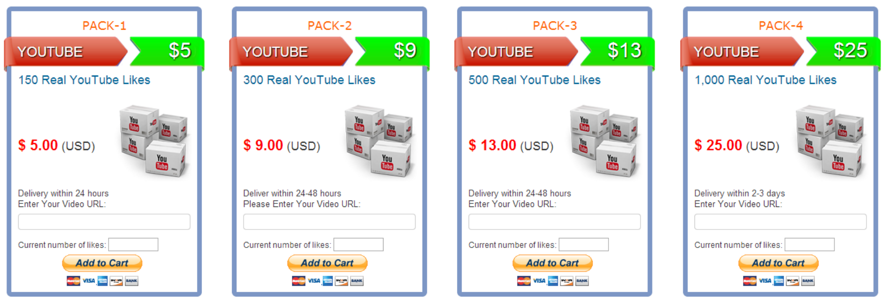
[www.buyfollowz.org]

[buymorelikes.com]

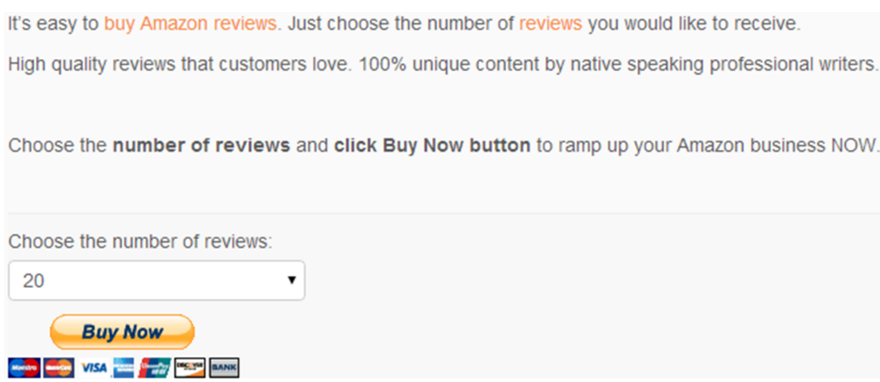


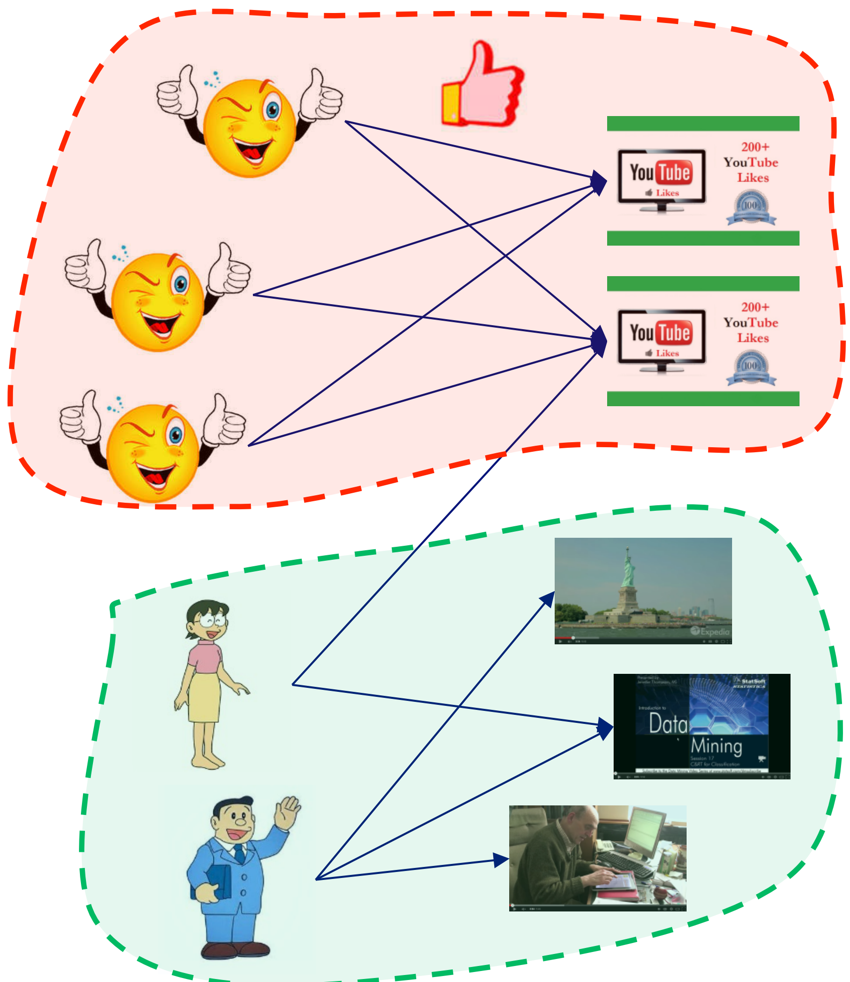
Fraud Detection: Graph Analysis Problem

[buycheaplikes.com]



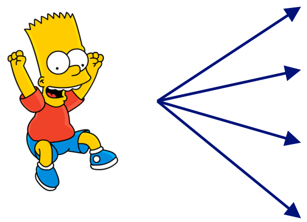
[reviewsteria.com]





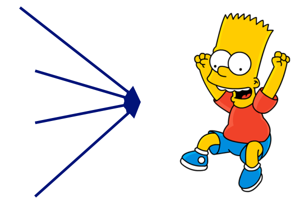
Carnegie Mellon

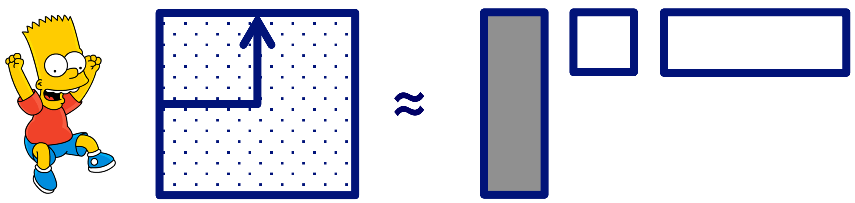
Behavior-based Features

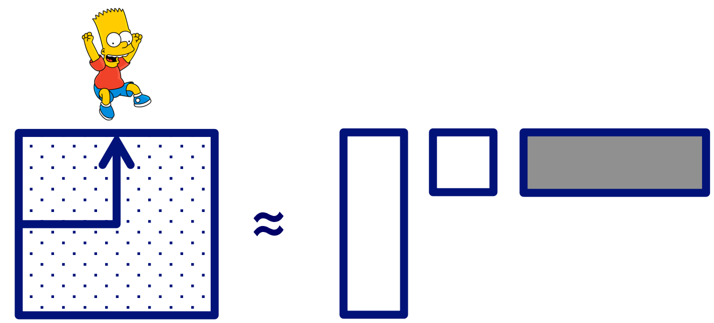


Follower behavior

Followee behavior







Out-degree

1st left singular vector (Hubness)

2nd left singular vector

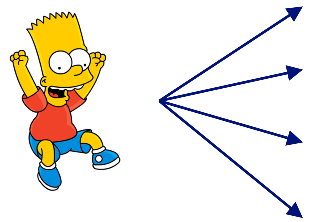
In-degree

1st right singular vector (Authoritativeness)

2nd right singular vector

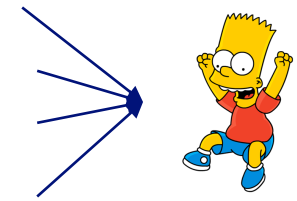
Carnegie Mellon

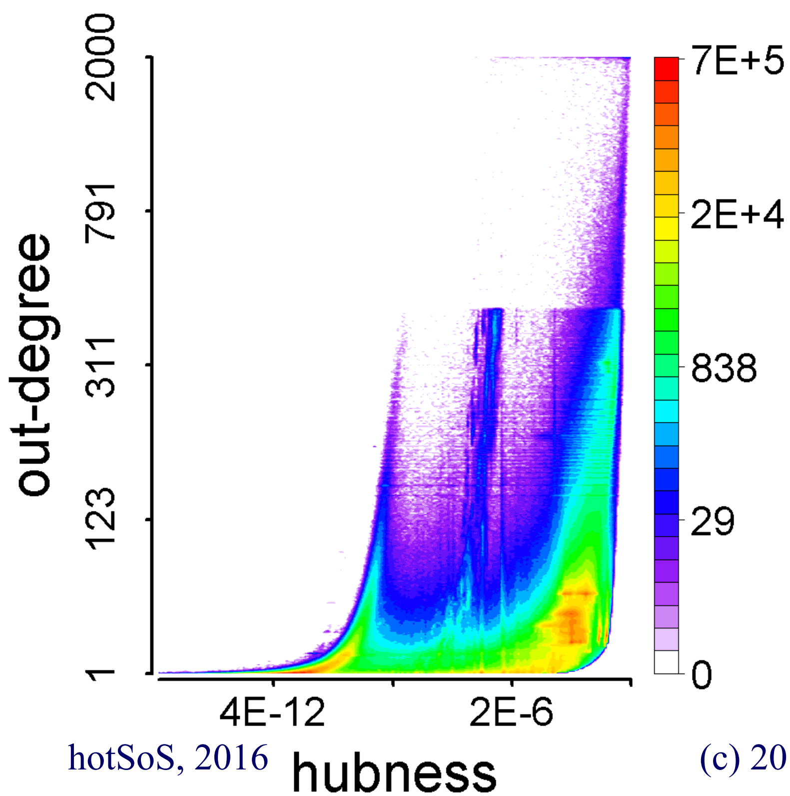
Behavior-based Feature Space

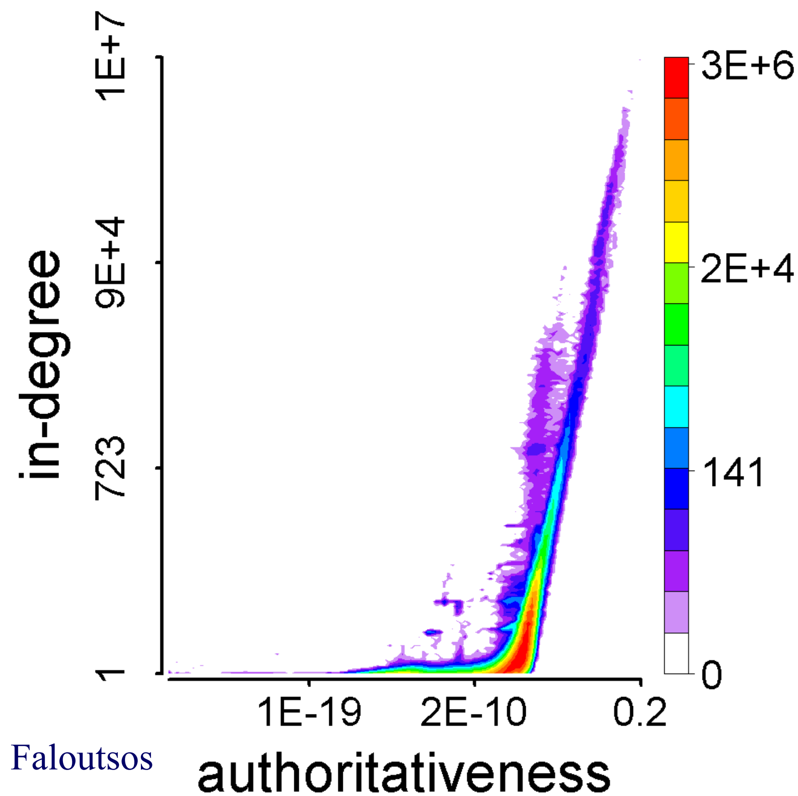


Follower behavior

Followee behavior

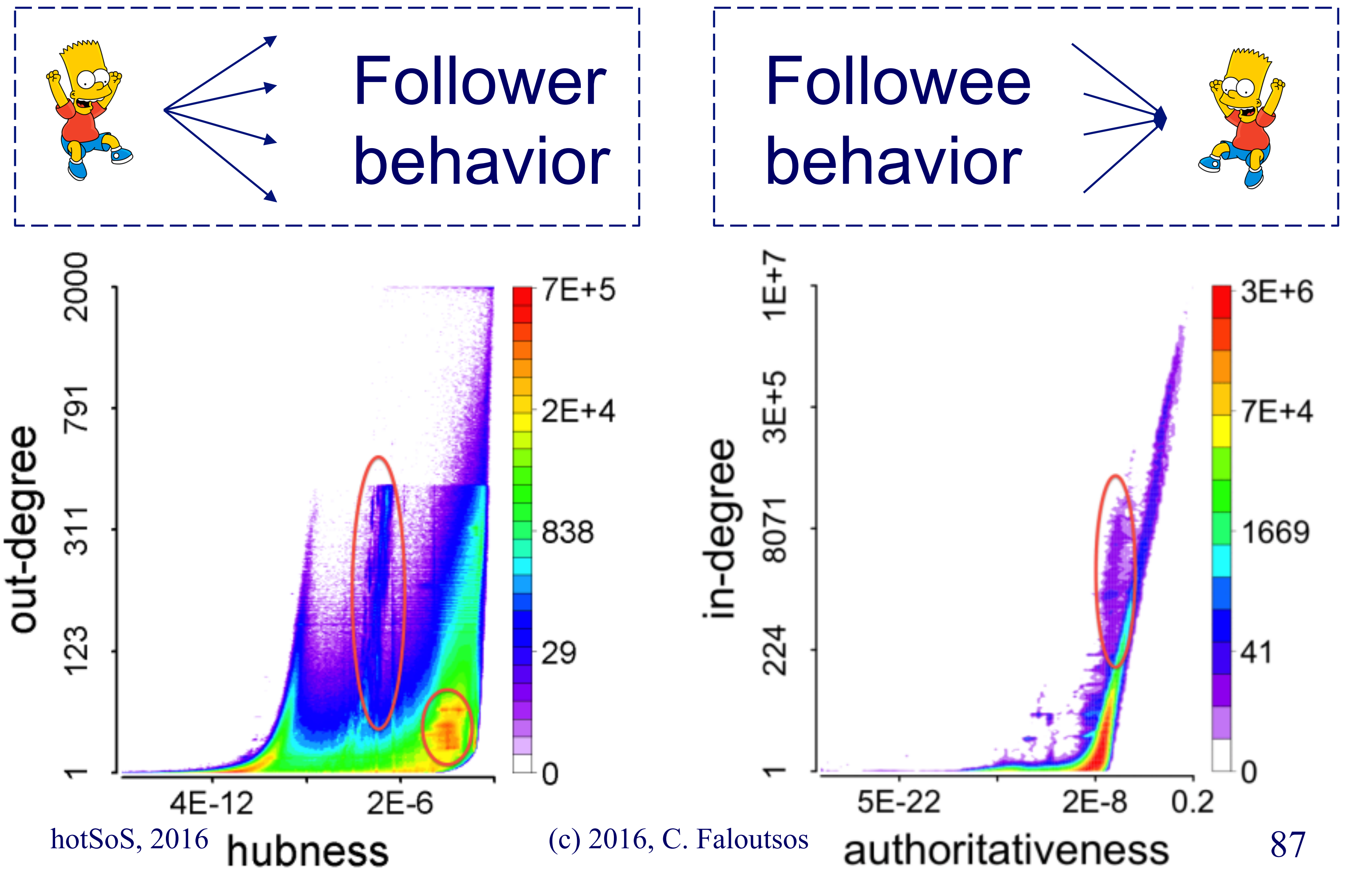




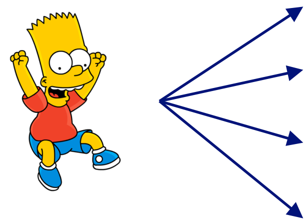


Carnegie Mellon

Before CatchSync

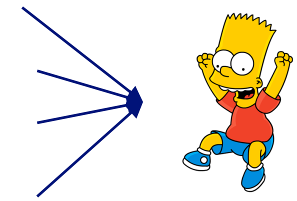


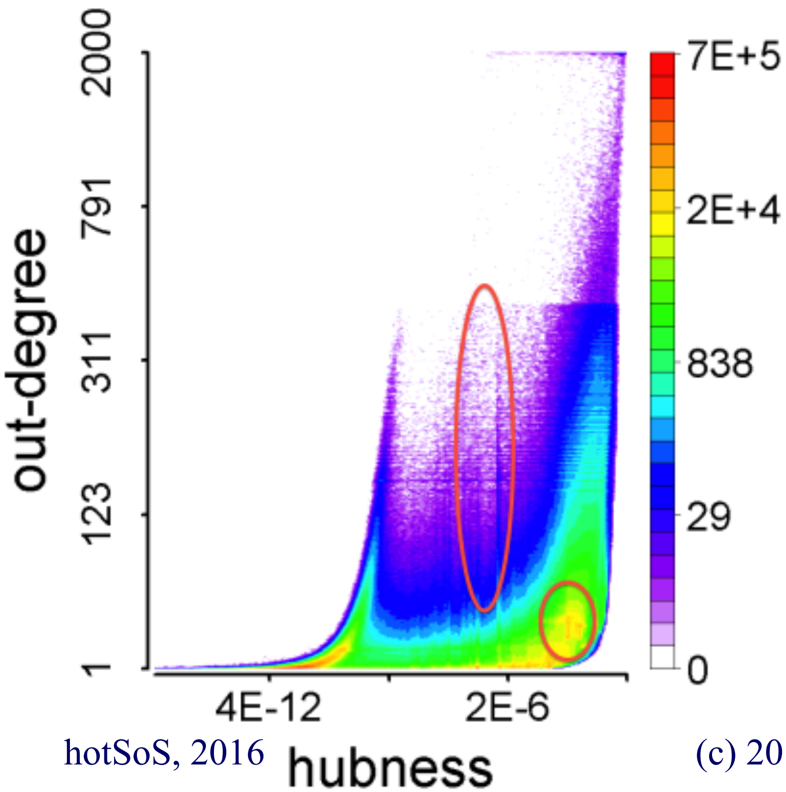
After CatchSync

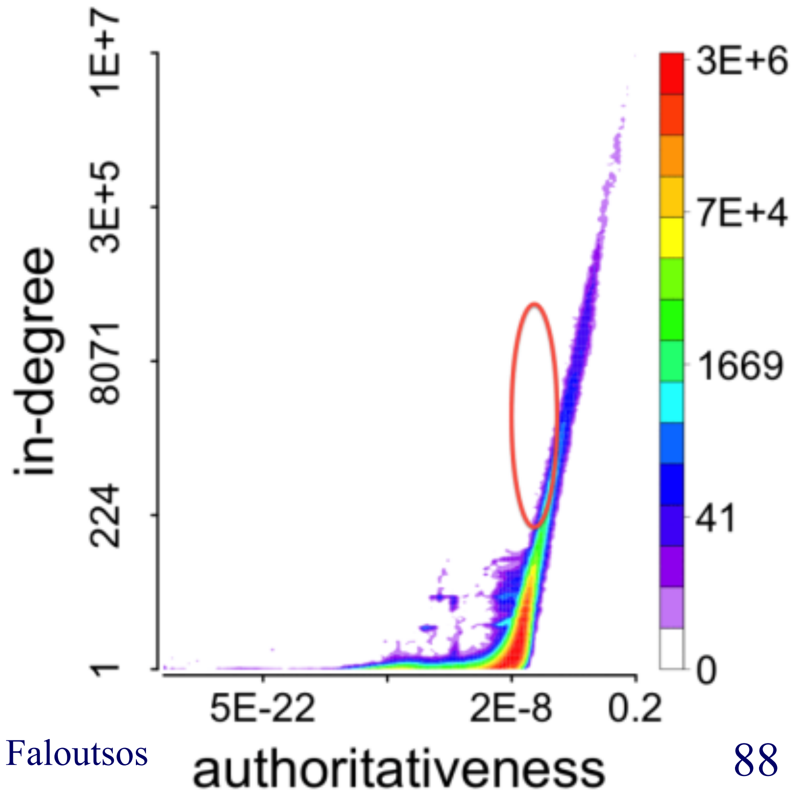


Follower behavior

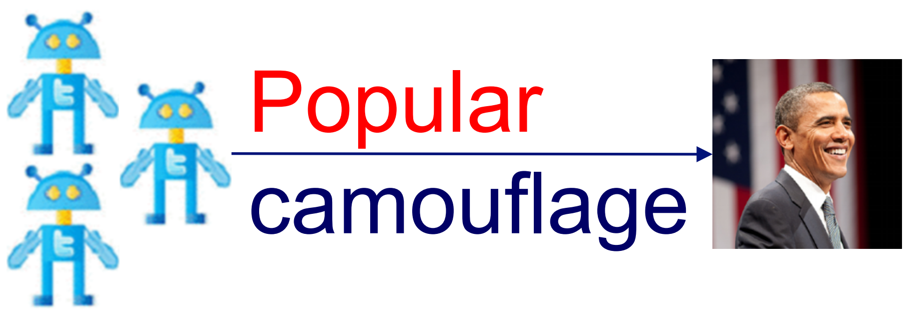
Followee behavior

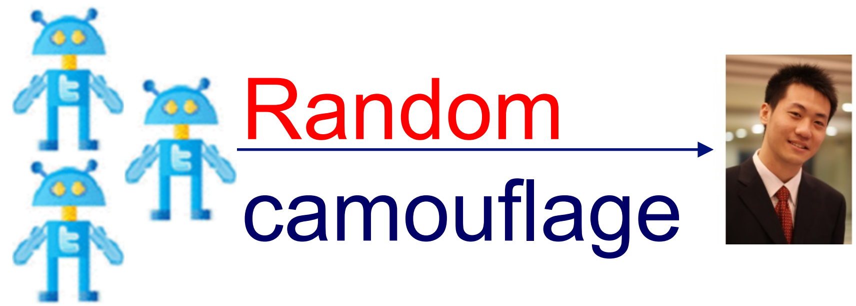


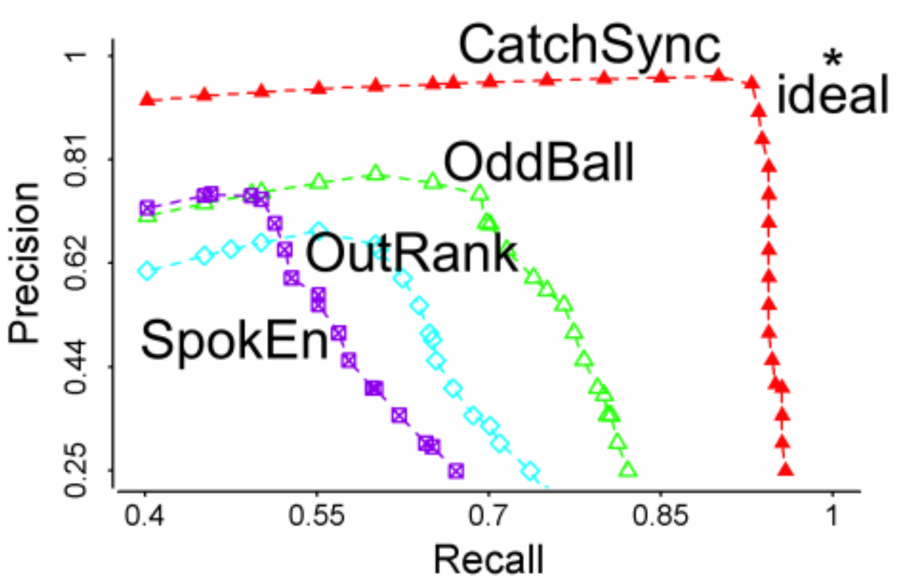


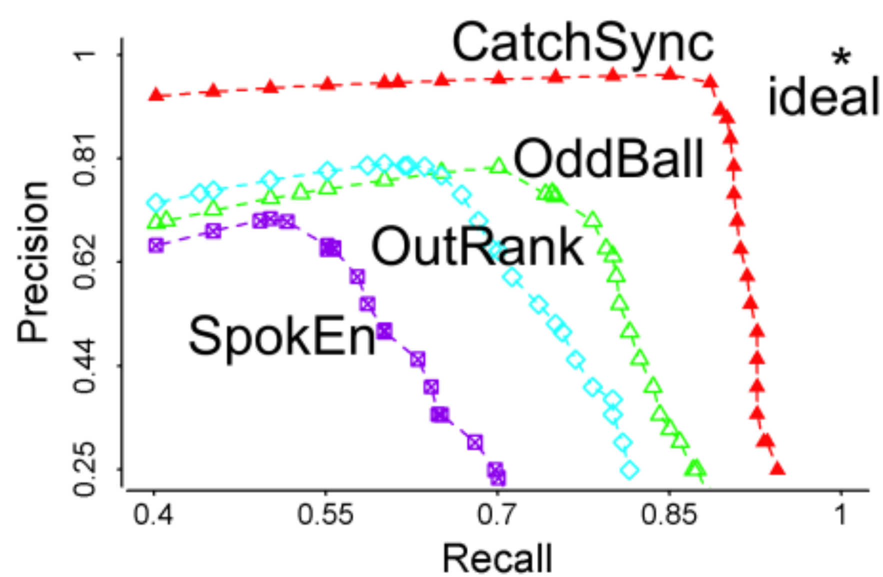


Q3: Is CatchSync Robust?



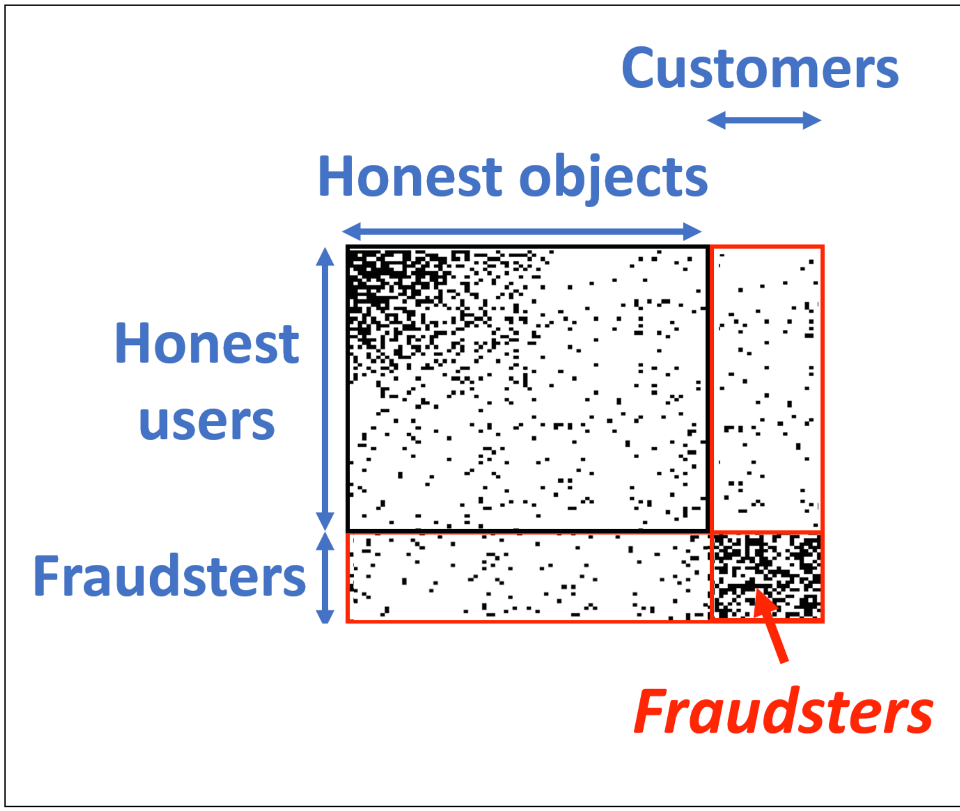






Detecting Review Spam

Many existing methods detect fraudsters using dense subgraph detection.



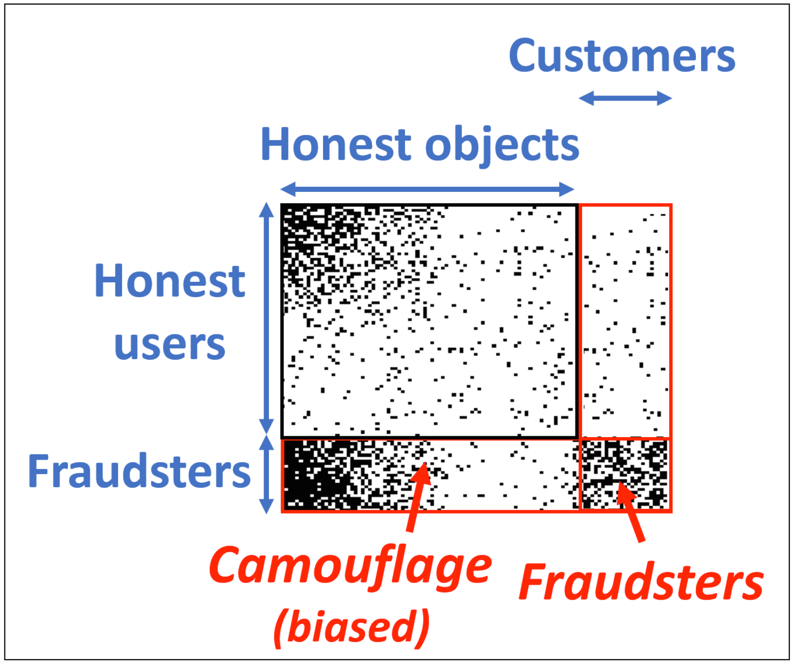
Spectral methods

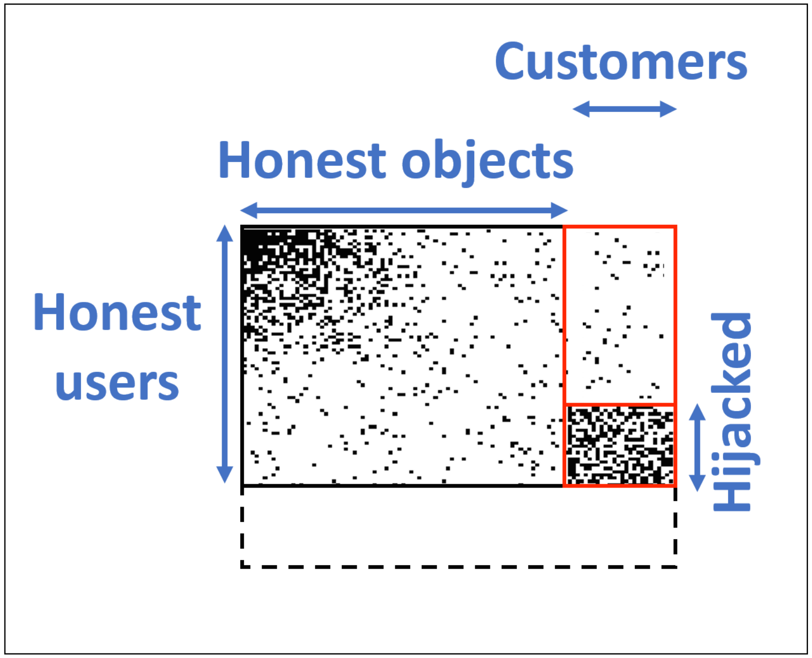
Clustering

Belief propagation

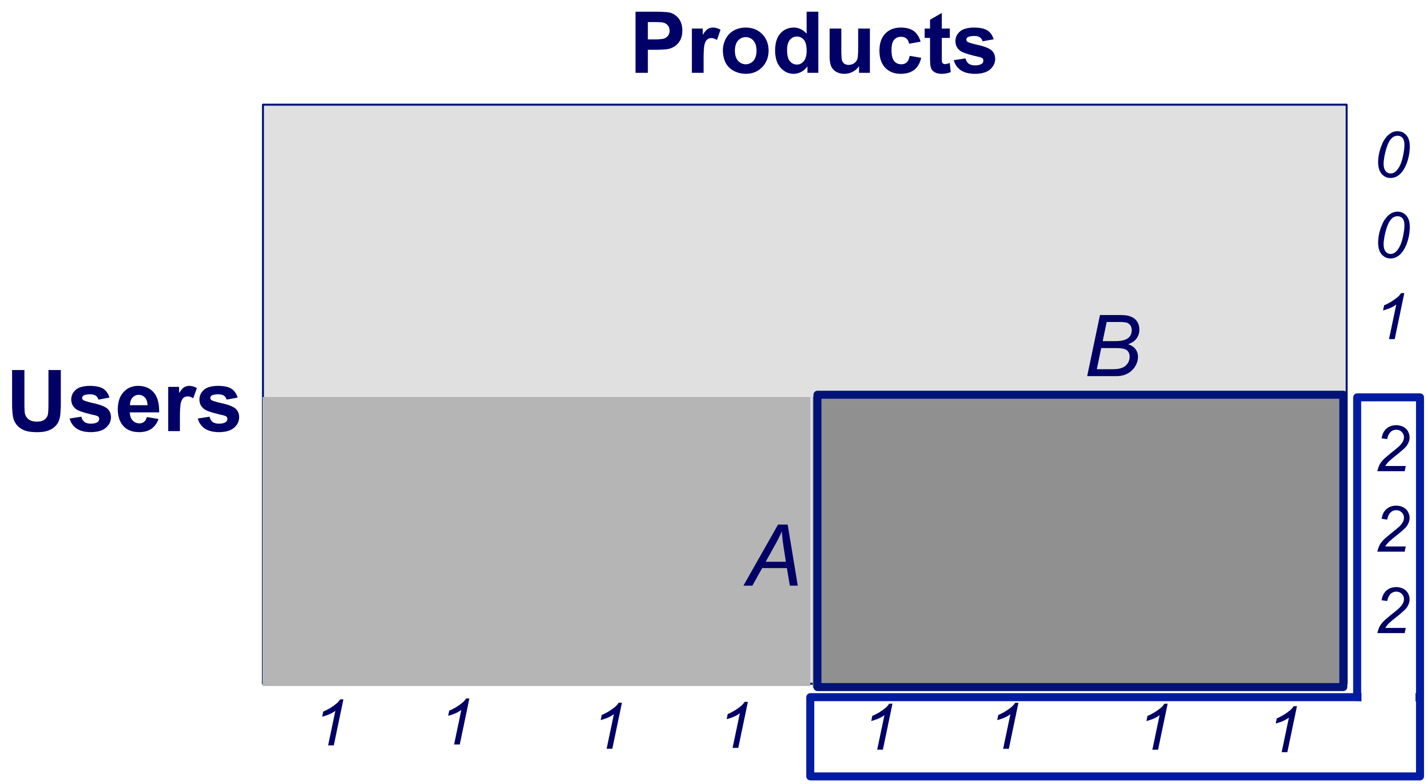
Evading Detection

Attackers can evade detection using *camouflage*.



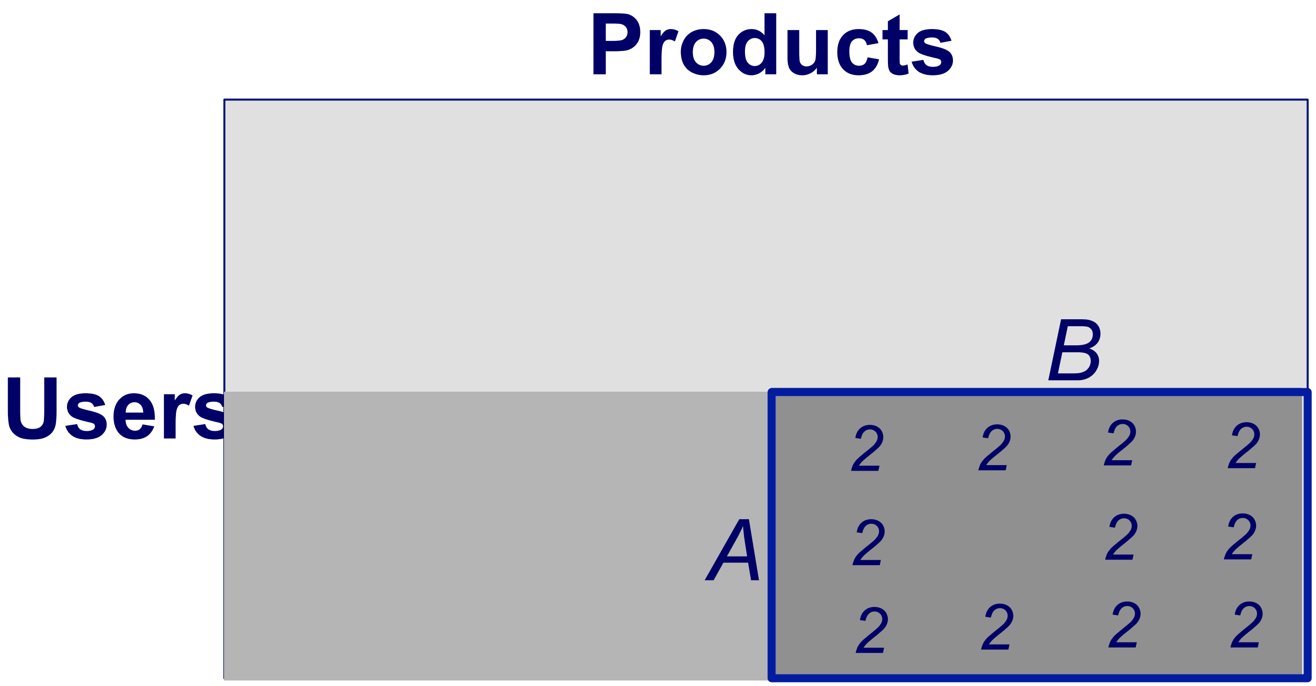


Node suspiciousness



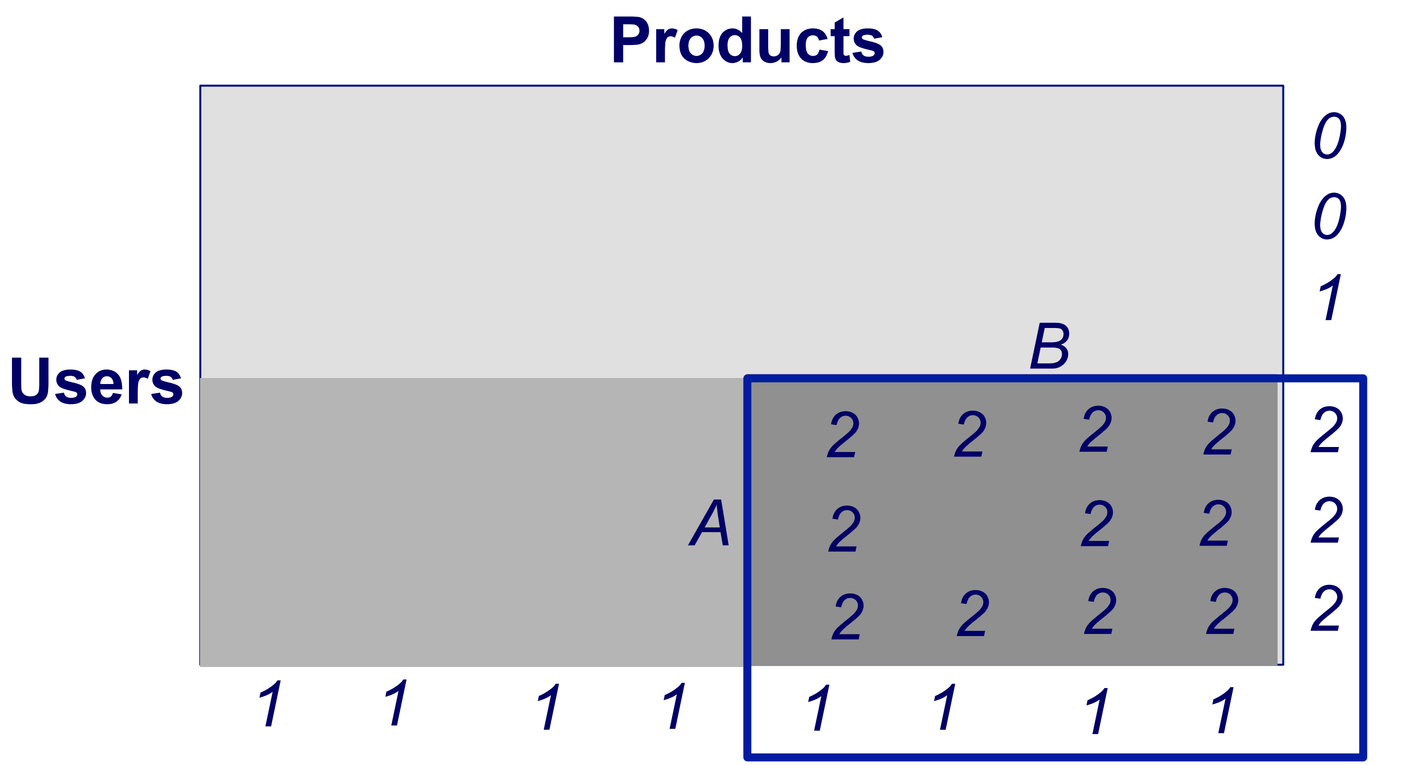
Node suspiciousness of (A,B) = 10

Edge suspiciousness



Edge suspiciousness of (A,B)
= 22

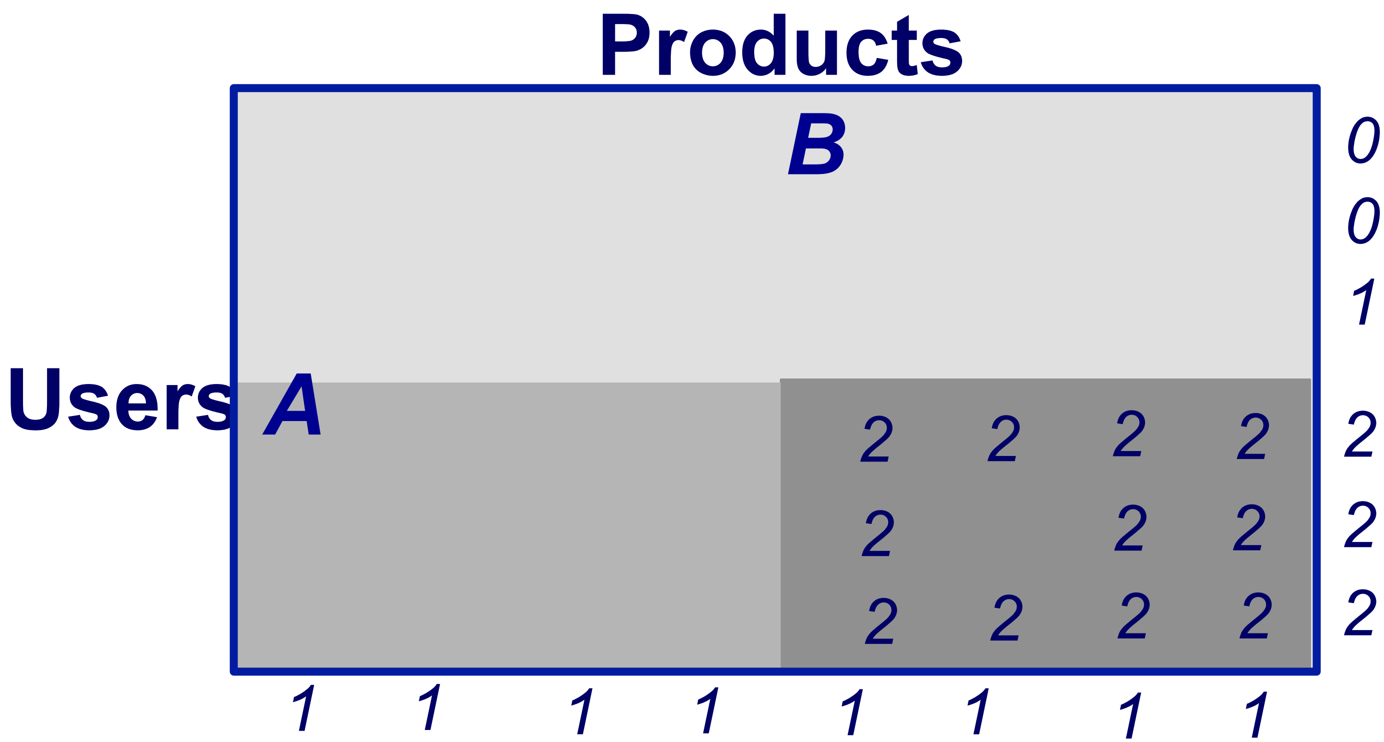
Total suspiciousness



$f(A,B) = 32$

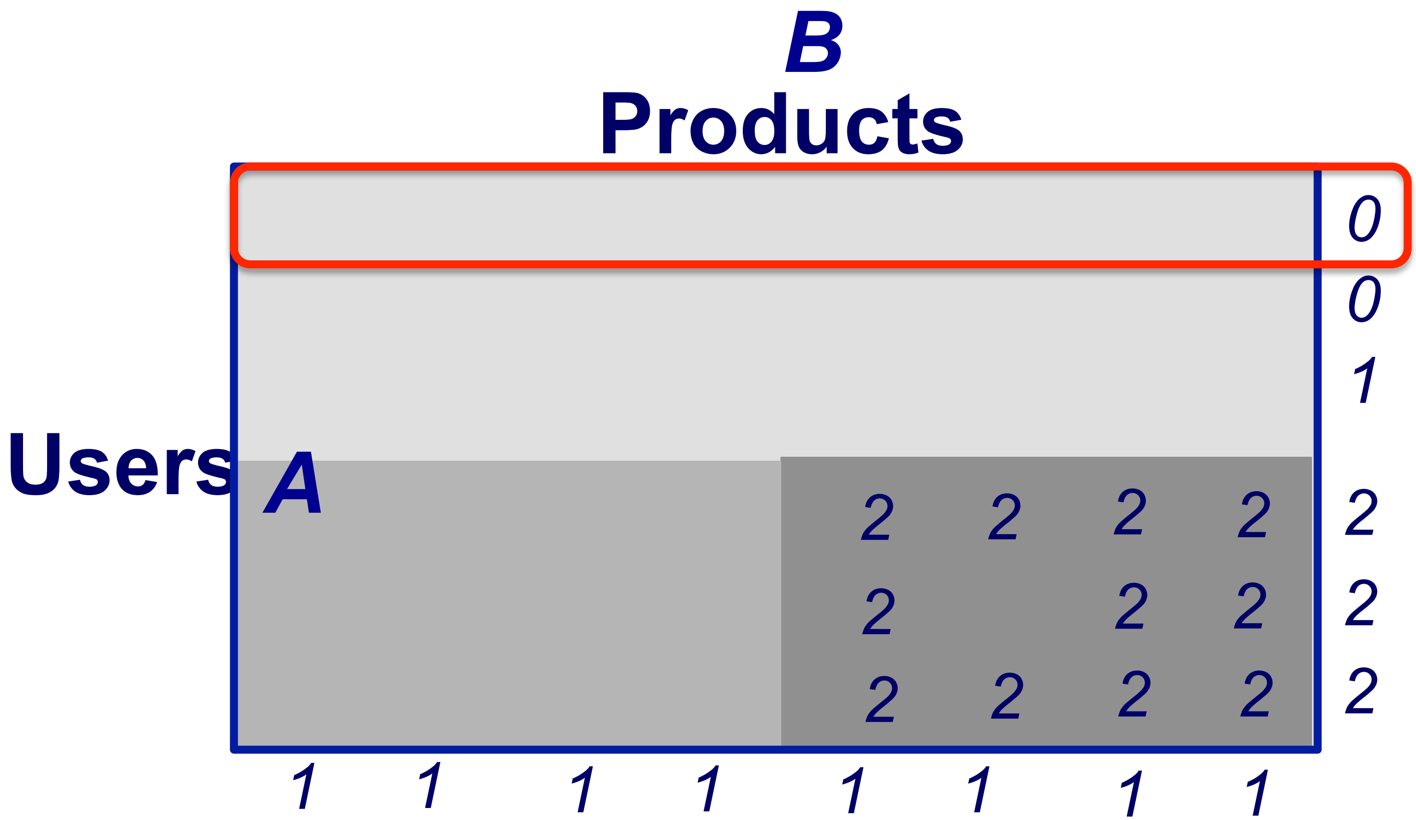
$\boldsymbol{f(A,B)}$ = (edge susp.) + (node susp.)

Greedy Algorithm



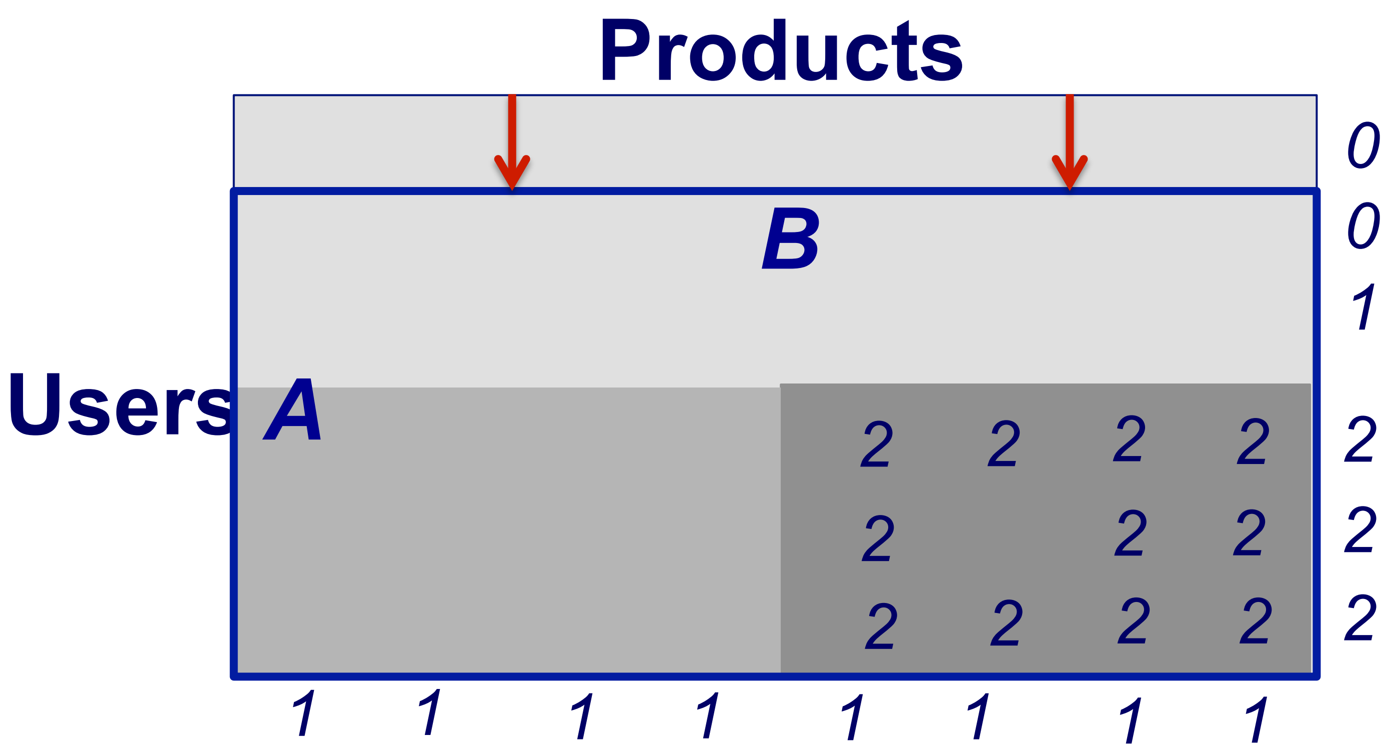
Start with A, B as all users / products

Greedy Algorithm



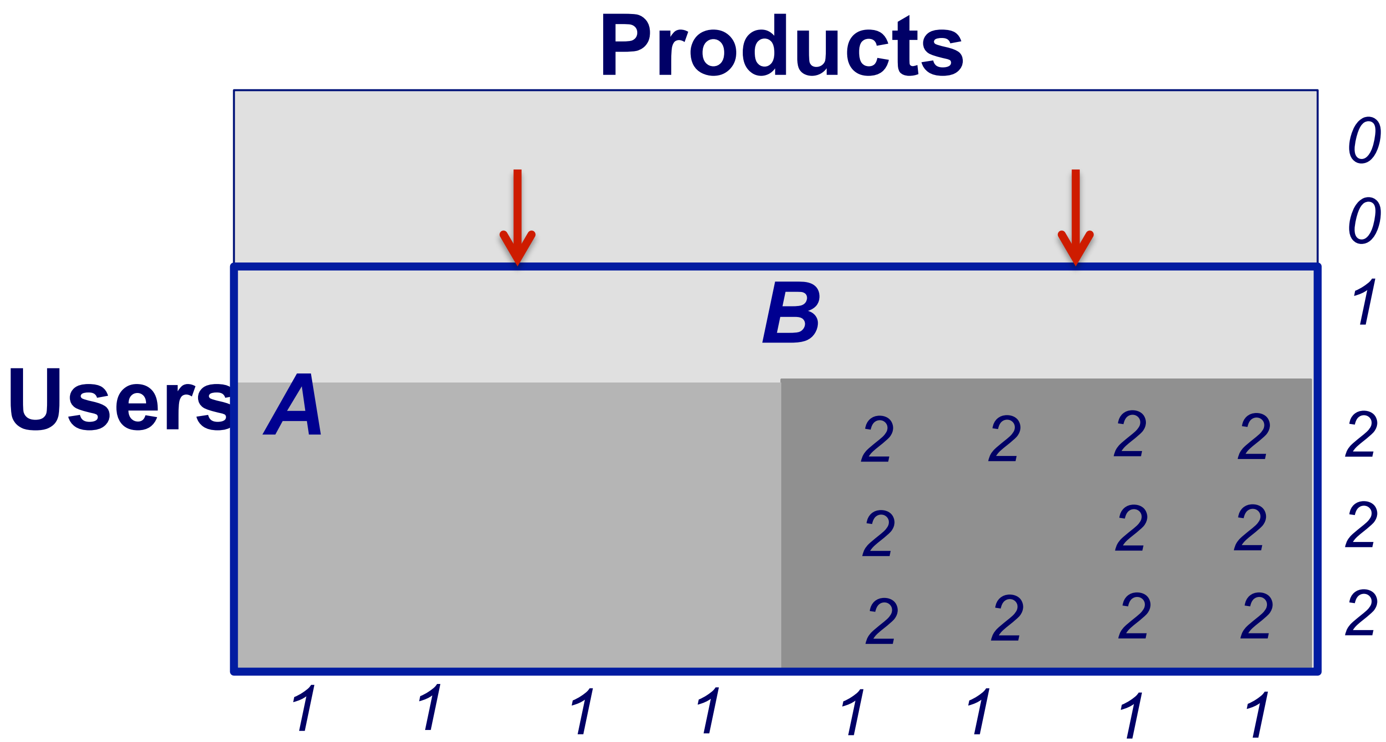
Delete rows / columns greedily to maximize g

Greedy Algorithm



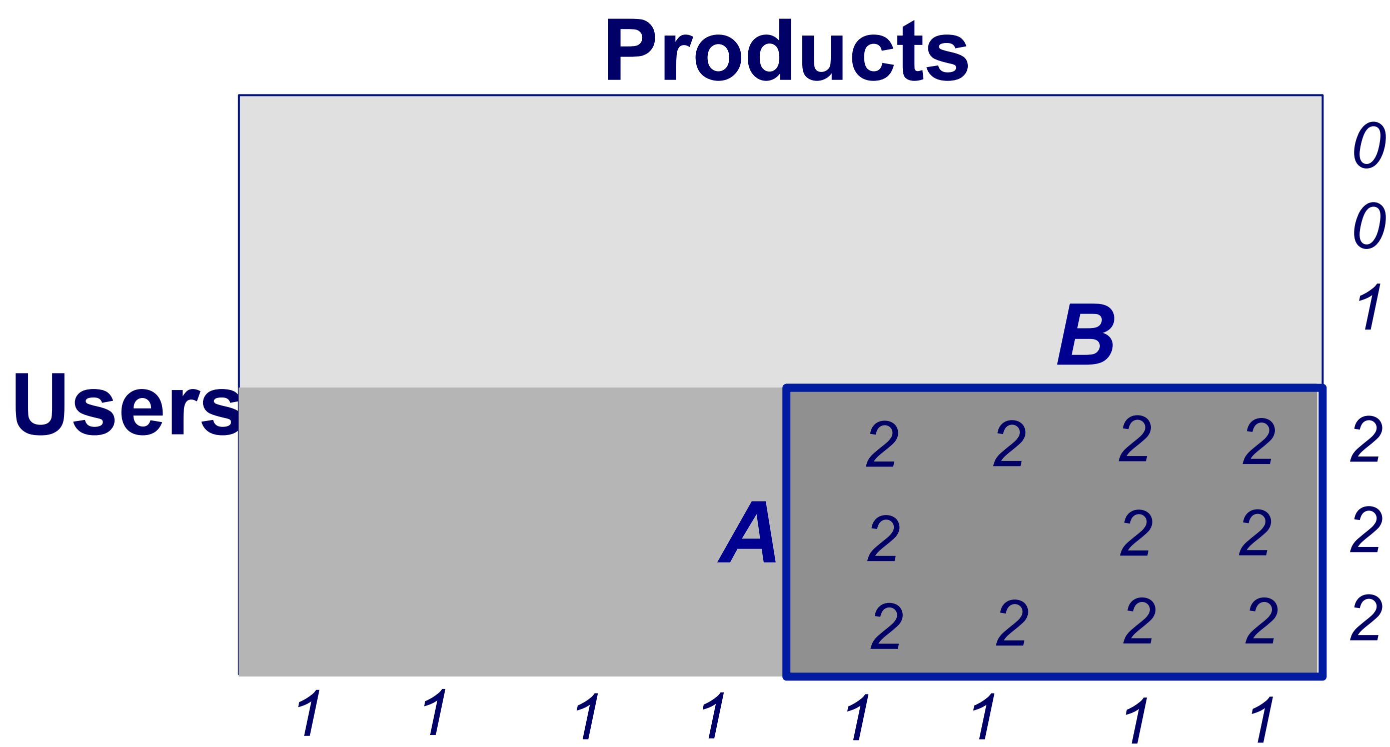
Delete rows / columns greedily to maximize g

Greedy Algorithm



Delete rows / columns greedily to maximize g

Greedy Algorithm

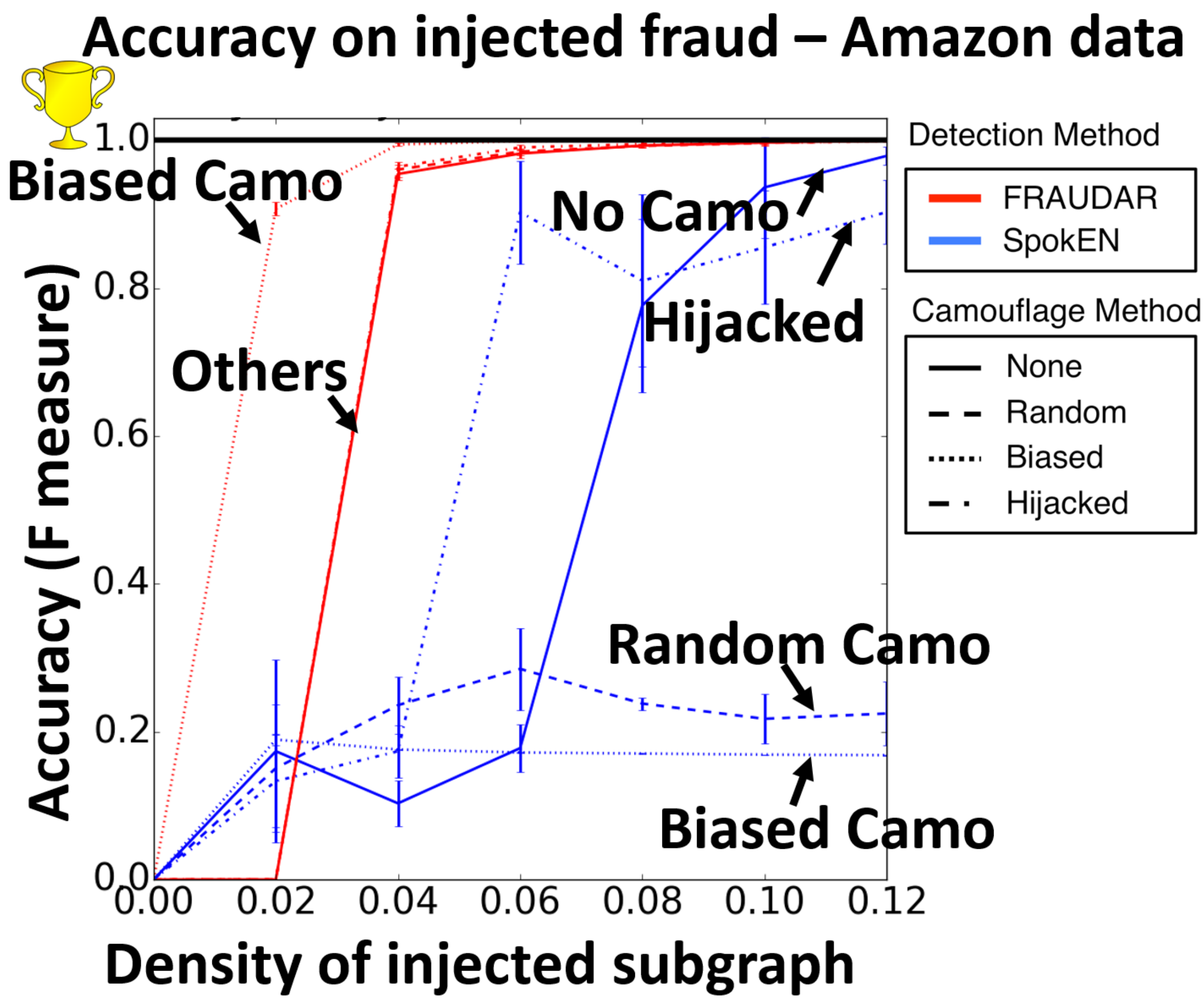


Delete rows / columns greedily to maximize g

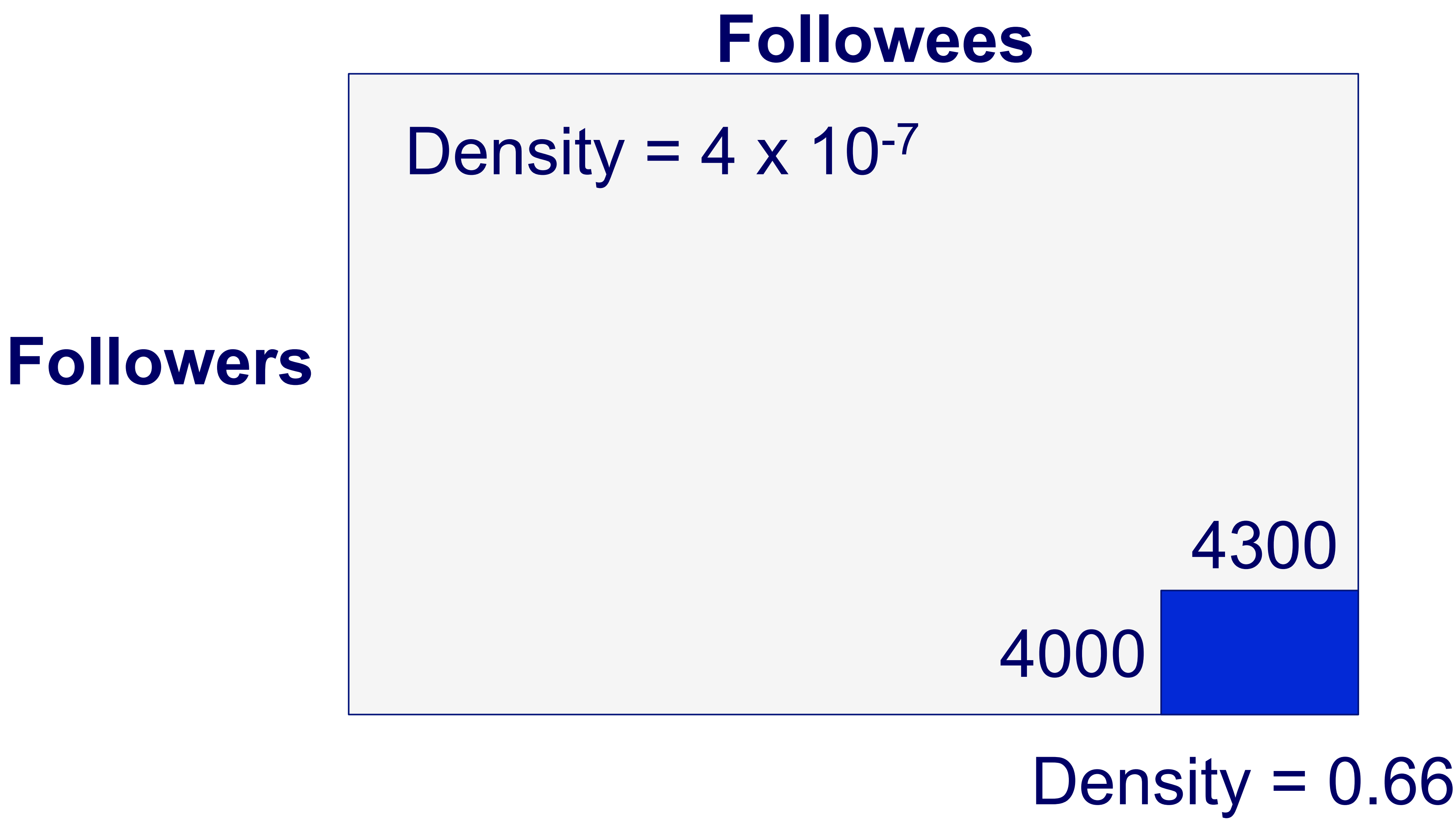
Experiments: Amazon data

- 24K x 4K Amazon review graph
- Injected dense blocks with various types of camouflage
  - None
  - Random camouflage
  - Biased camouflage
  - Hijacked accounts

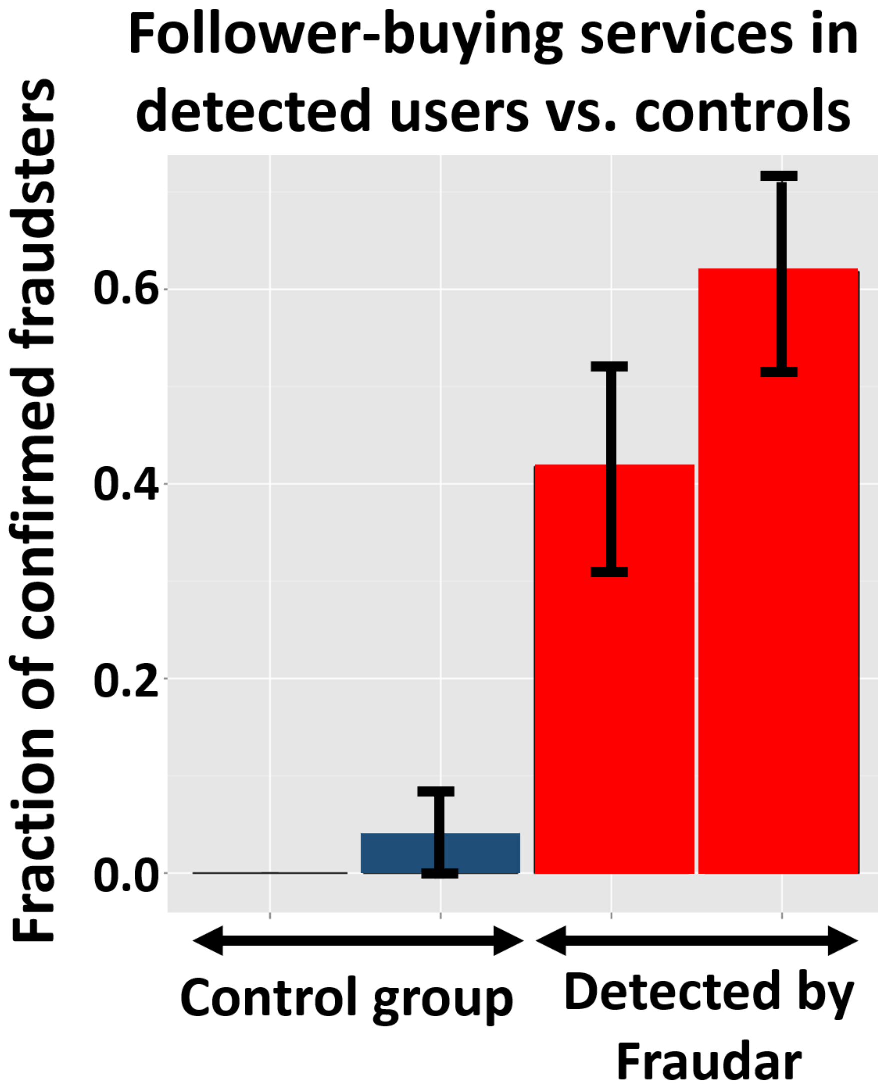
Experiments: Amazon



Twitter data

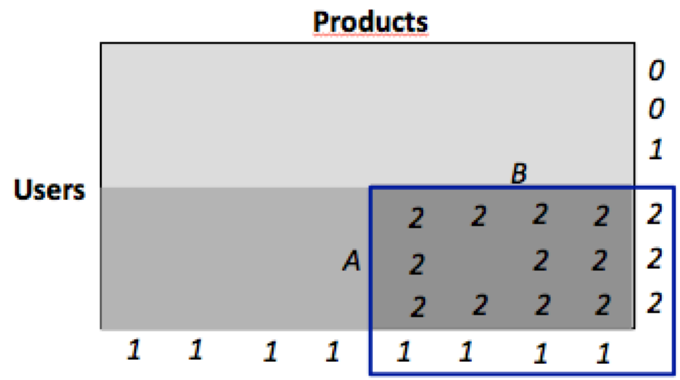


Twitter data



Conclusion

- **Average suspiciousness metric**



$$\boldsymbol{g(A,B)} = f(A,B) \,/\, (|A| + |B|)$$

- **Theoretical guarantees**

$$g(\mathcal{A} \cup \mathcal{B}) \geq \frac{1}{2} g_{OPT}$$

- **Effectiveness**