Carnegie Mellon

Anomaly detection in large graphs

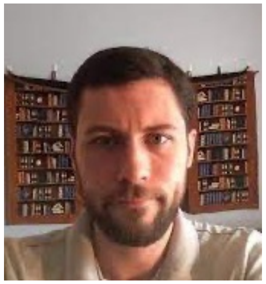
Christos Faloutsos

CMU

Thank you!

- Badel Mbanga

- Clifton Denning



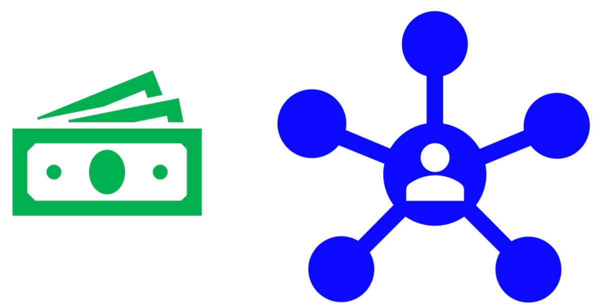
- Matthew Berezo

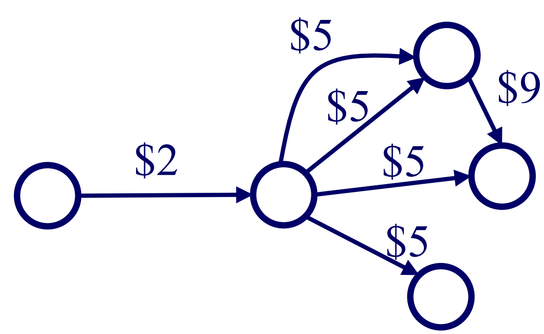
Roadmap

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

Graphs – why should we care?

PNC



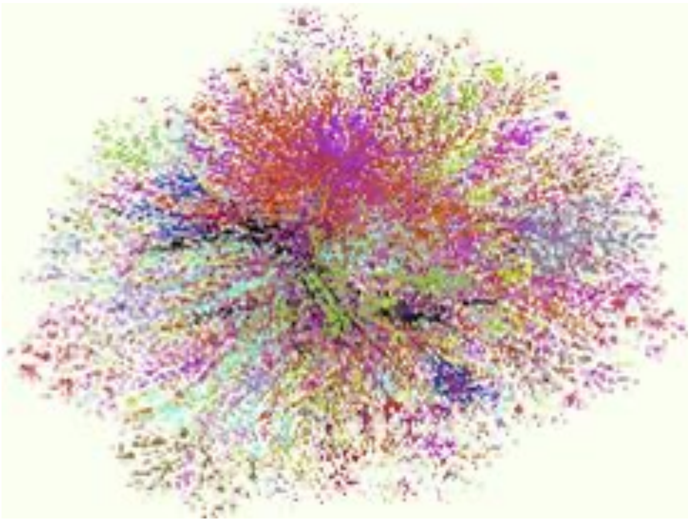


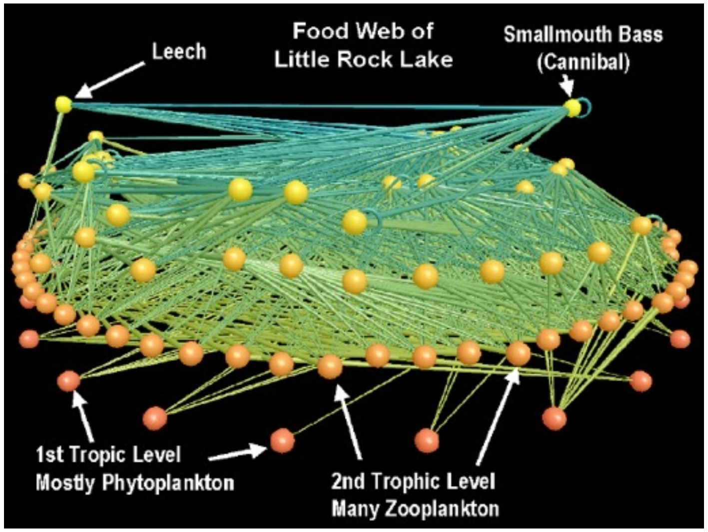
(source, destination, timestamp, $amount)

Graphs - why should we care?

>$10B; ~1B users

Graphs - why should we care?





Internet Map
[lumeta.com]

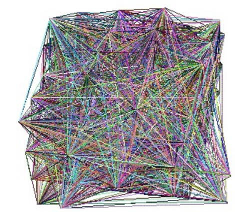
Food Web
[Martinez '91]

Graphs - why should we care?

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

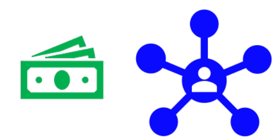
- Many-to-many db relationship -> graph

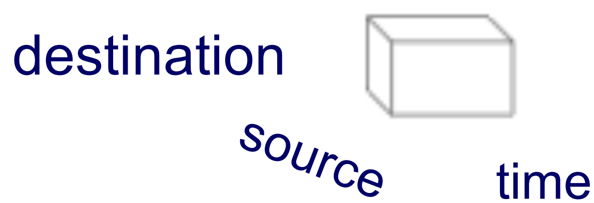
Motivating problems



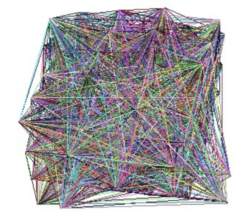
- P1: patterns? Fraud detection?

- P2: patterns in time-evolving graphs / tensors

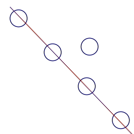




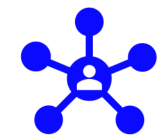
Motivating problems

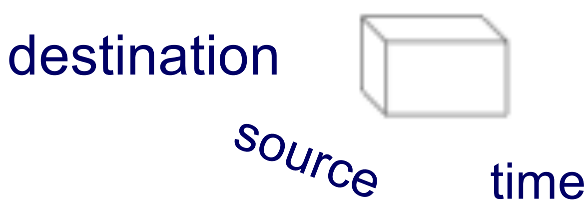


- P1: patterns? Fraud detection?

 Patterns  anomalies

- P2: patterns in time-evolving graphs / tensors

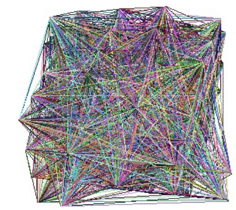
 



Roadmap

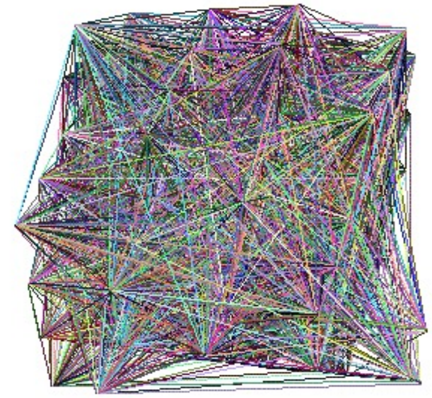
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns & fraud detection
- Part#2: time-evolving graphs; tensors
- Conclusions

Carnegie Mellon



Part 1: Patterns, & fraud detection

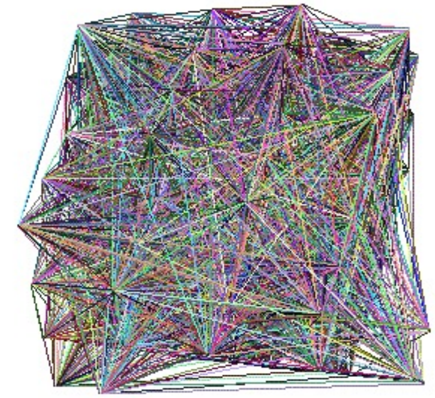
Laws and patterns



- Q1: Are real graphs random?

Carnegie Mellon

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

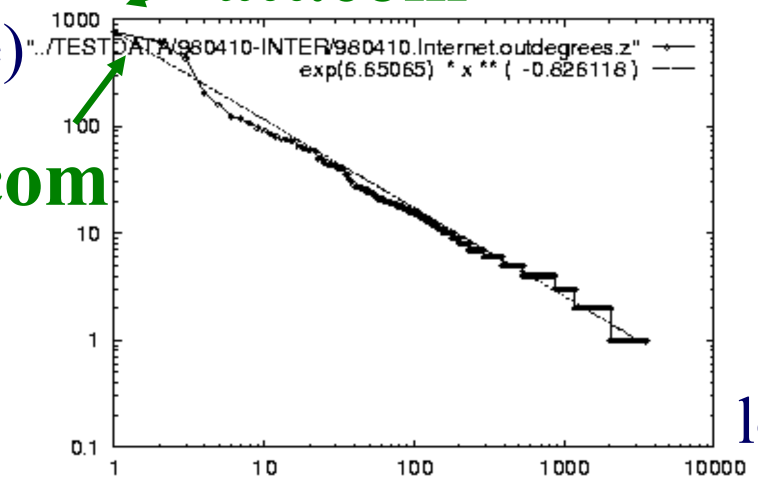
- Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

internet domains

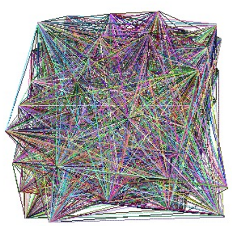
att.com

log(degree)

ibm.com



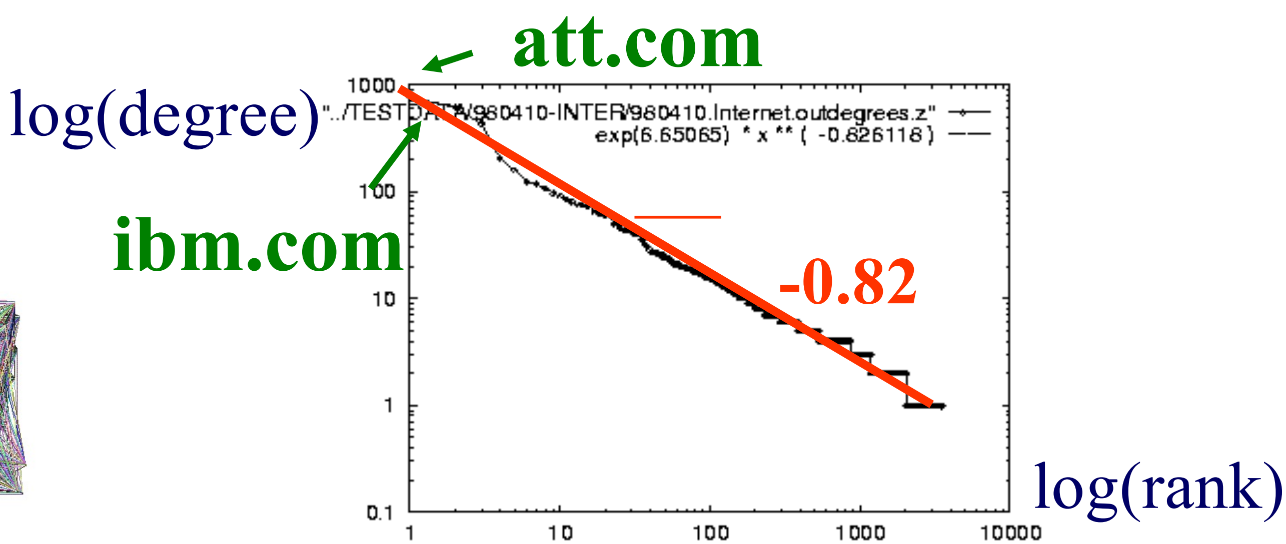
log(rank)

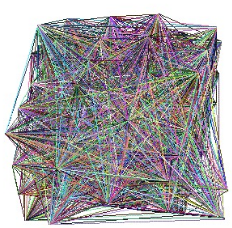


Solution# S.1

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

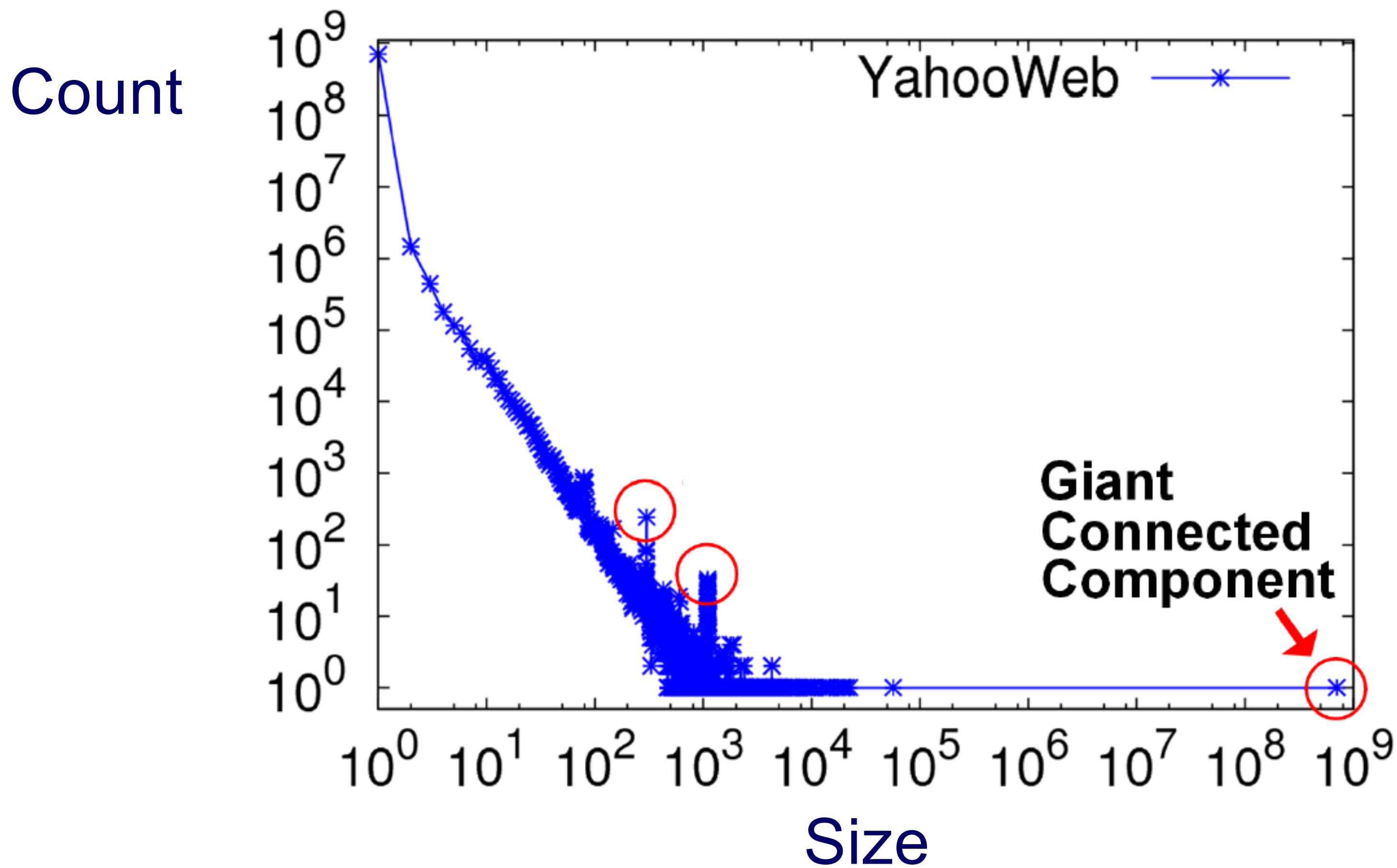
internet domains





S2: connected component sizes

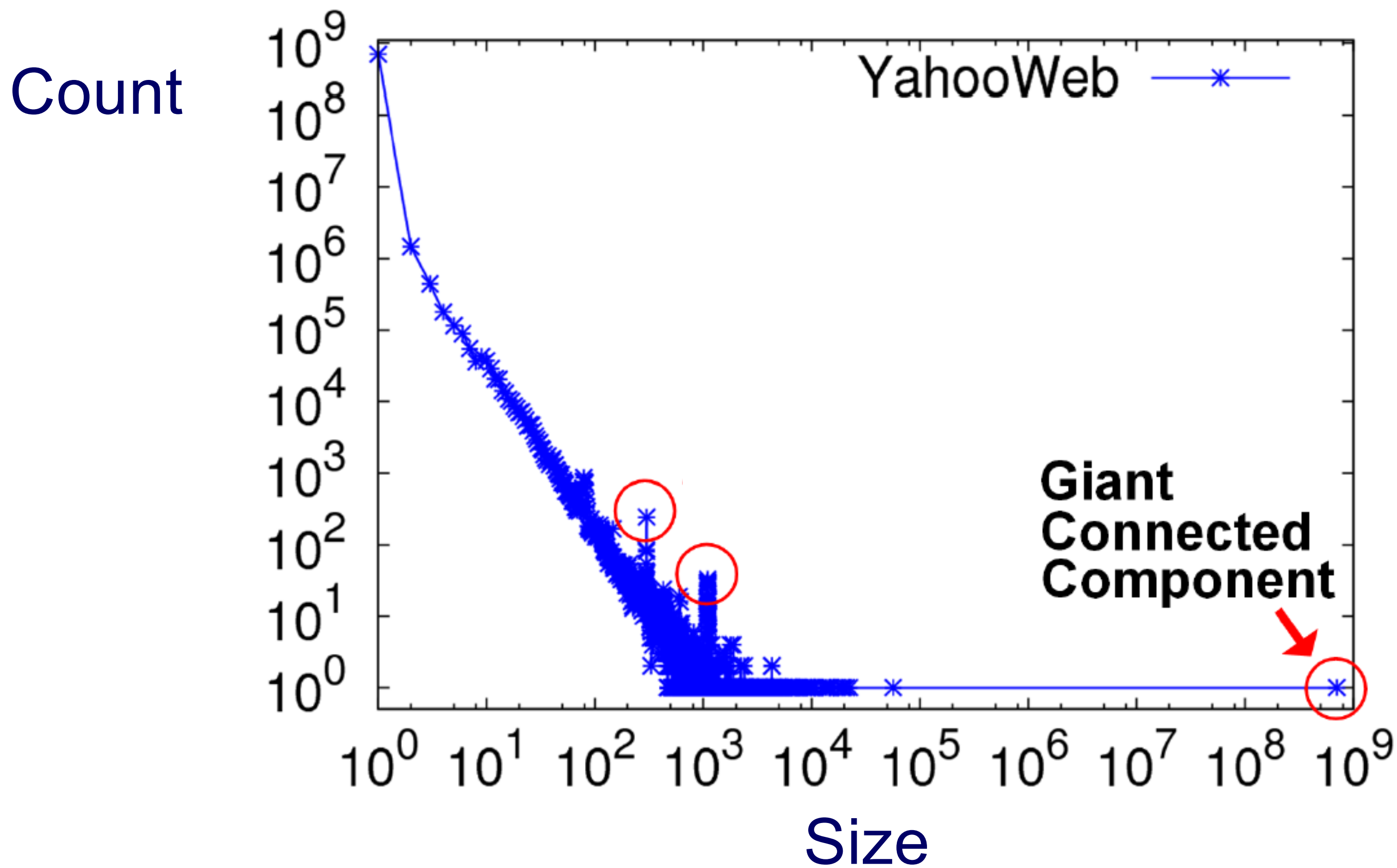
- Connected Components – 4 observations:



1.4B nodes
6B edges

S2: connected component sizes

- Connected Components

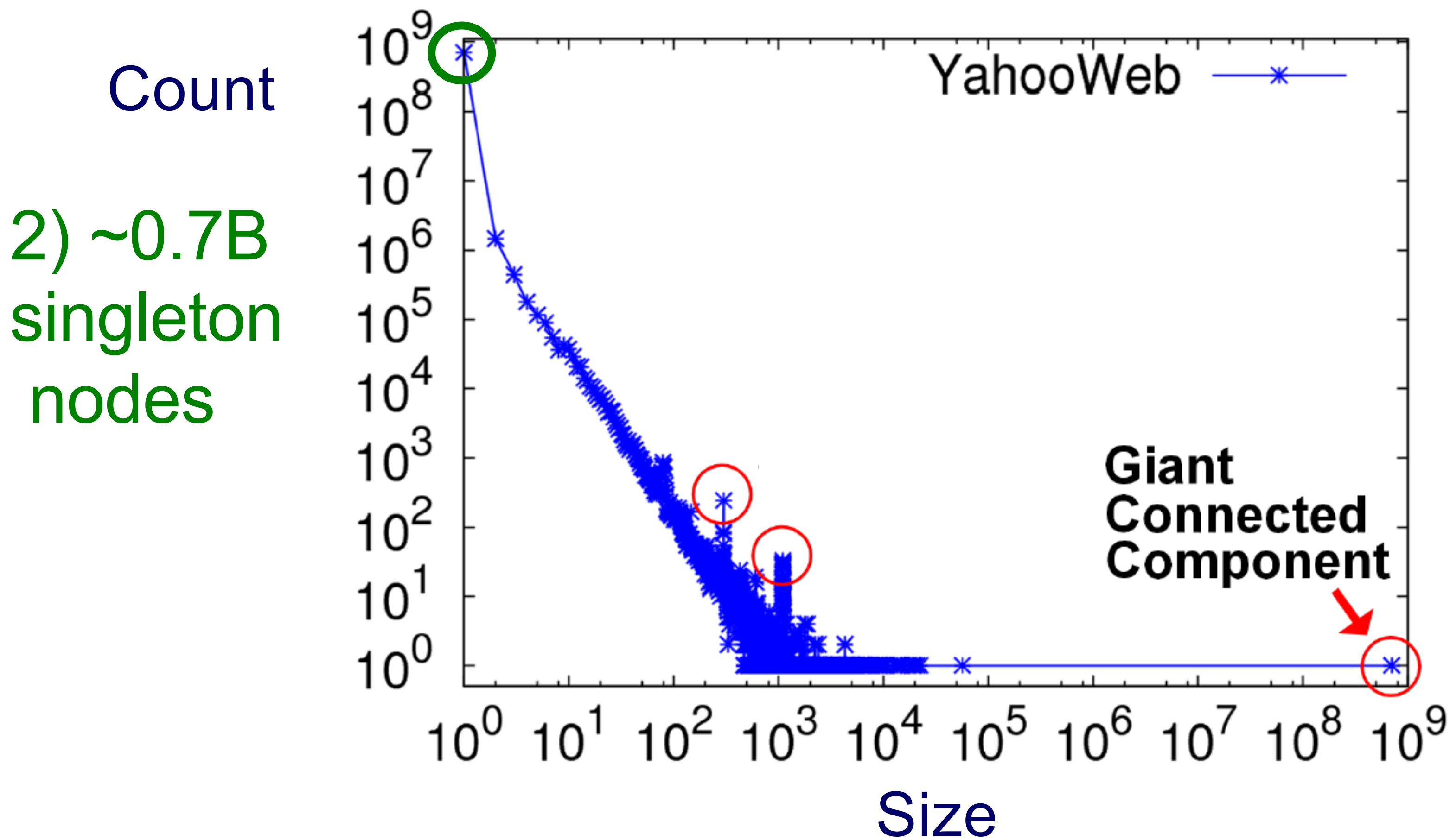


1) 10K x larger than next

Carnegie Mellon

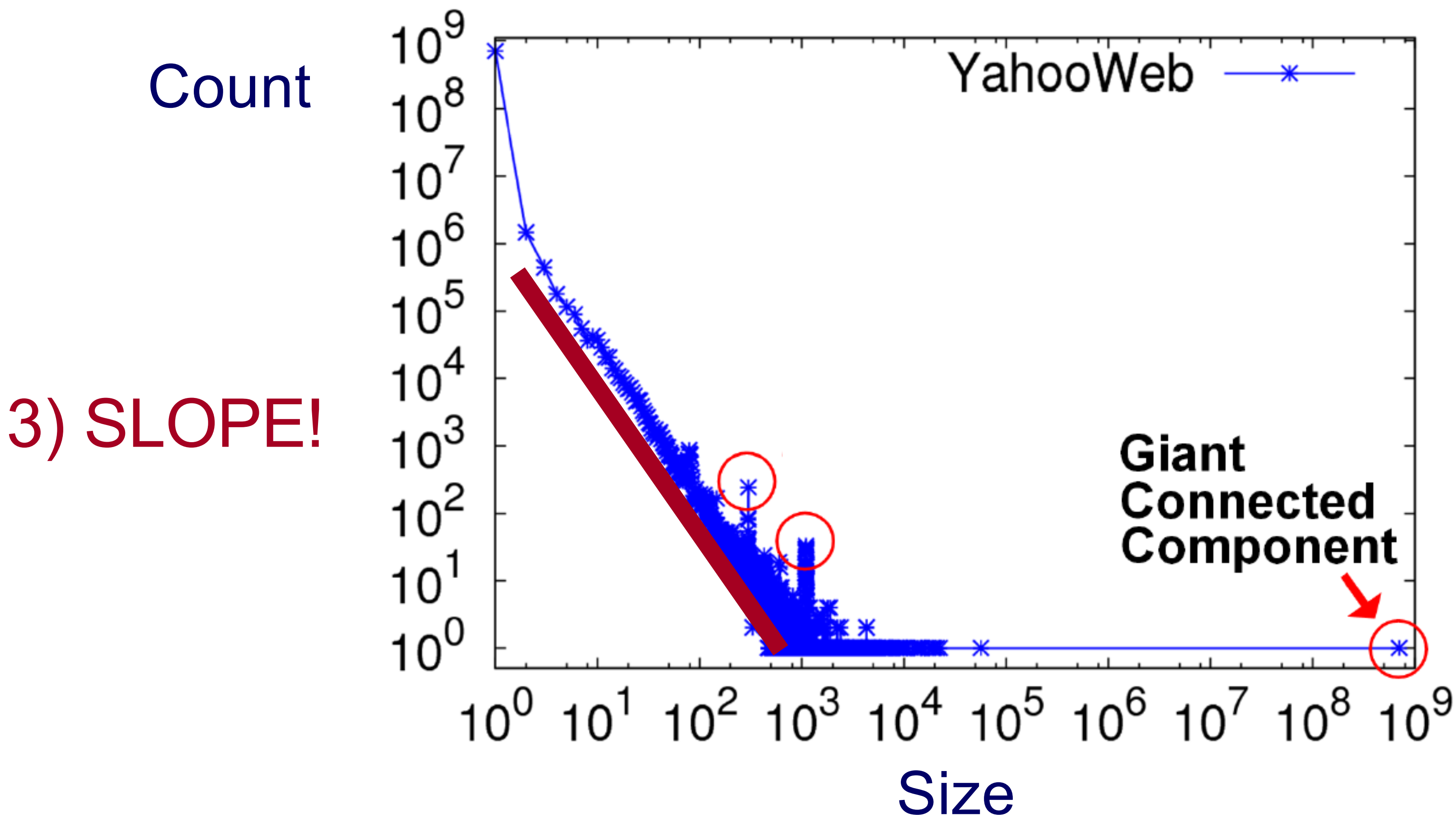
S2: connected component sizes

- Connected Components



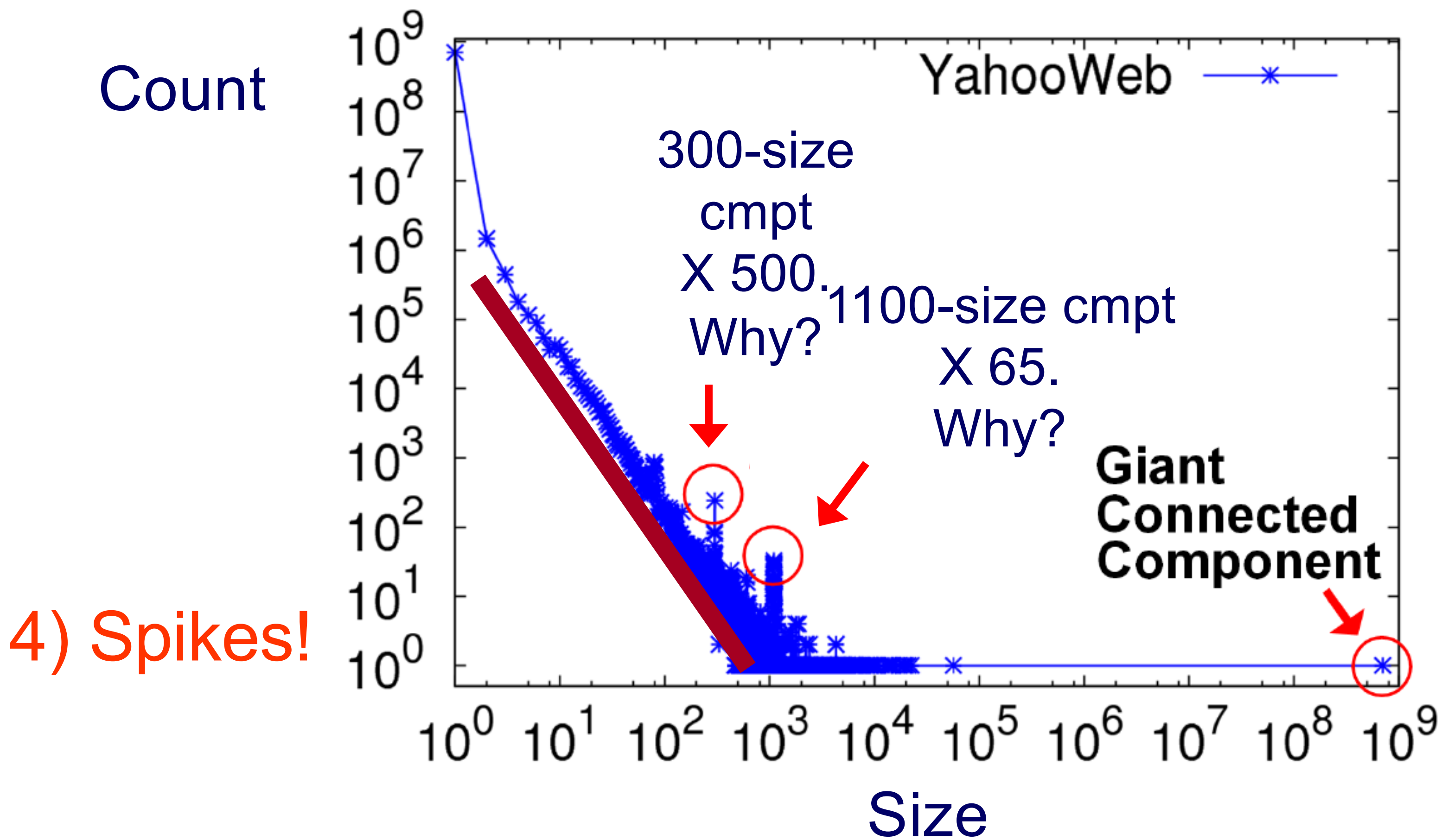
S2: connected component sizes

- Connected Components



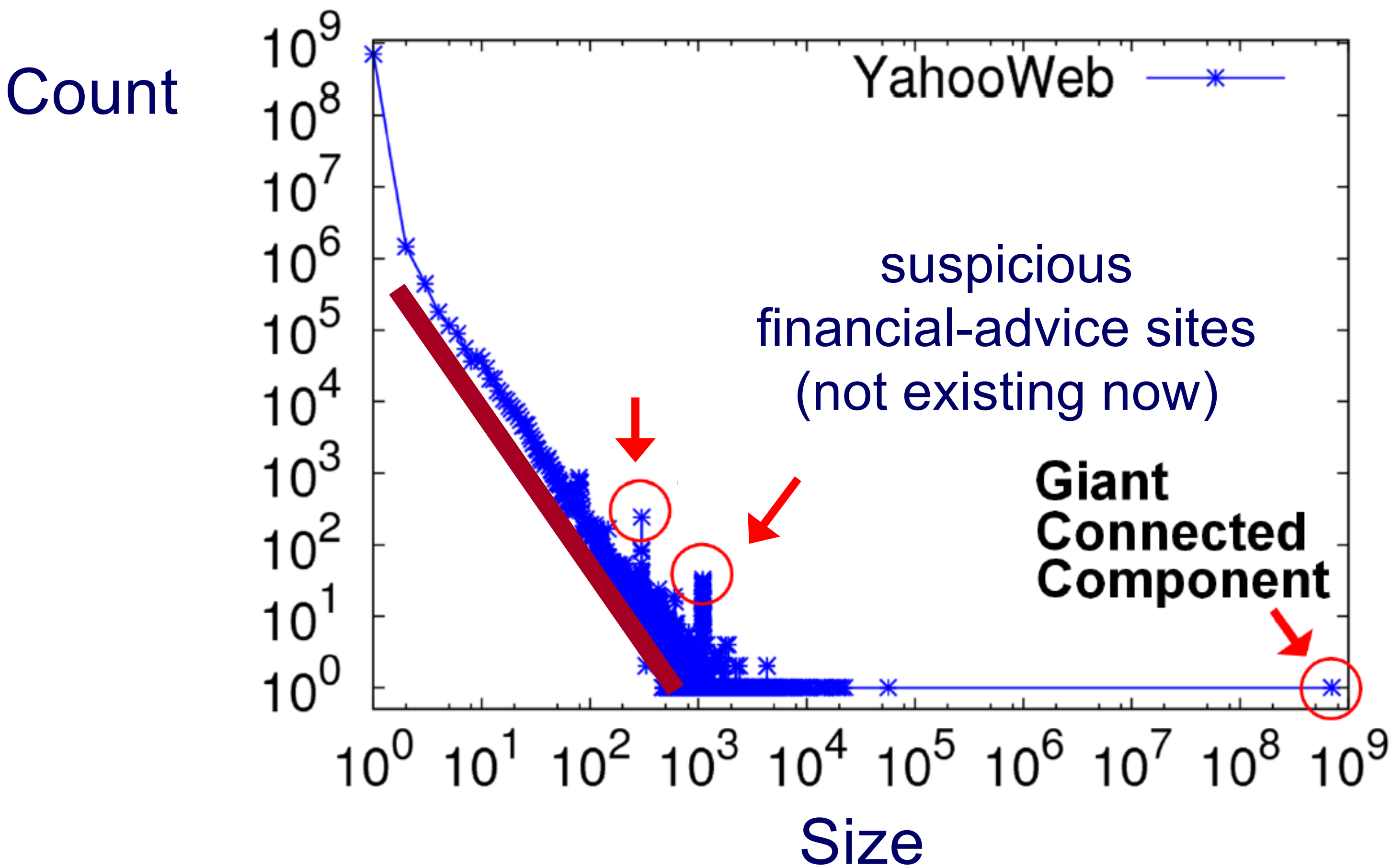
S2: connected component sizes

- Connected Components



S2: connected component sizes

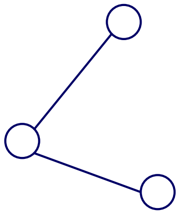
- Connected Components



Roadmap

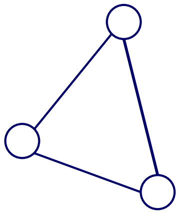
- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns: Degree; Triangles
  - P1.2: Anomaly/fraud detection
- Part#2: time-evolving graphs; tensors
- Conclusions

Solution# S.3: Triangle 'Laws'

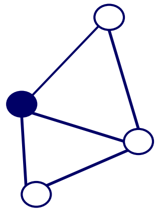


- Real social networks have a lot of triangles

Solution# S.3: Triangle 'Laws'

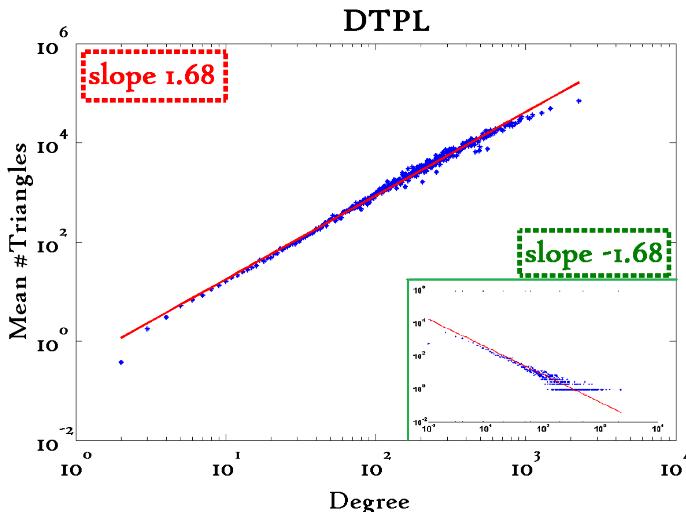


- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?

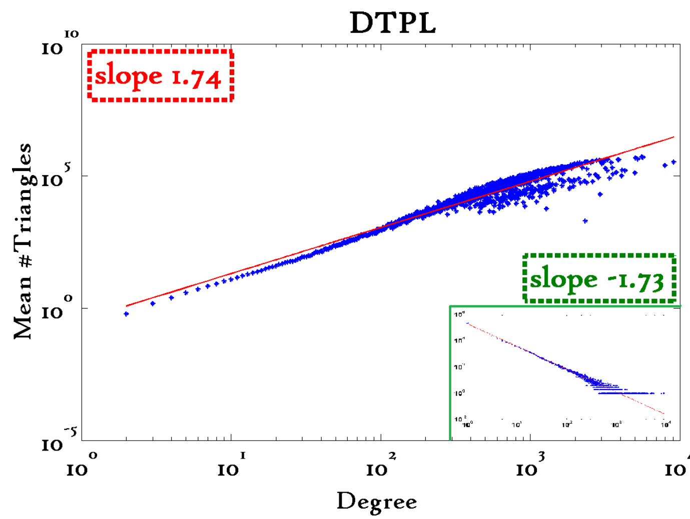


Triangle Law: #S.3
[Tsourakakis ICDM 2008]

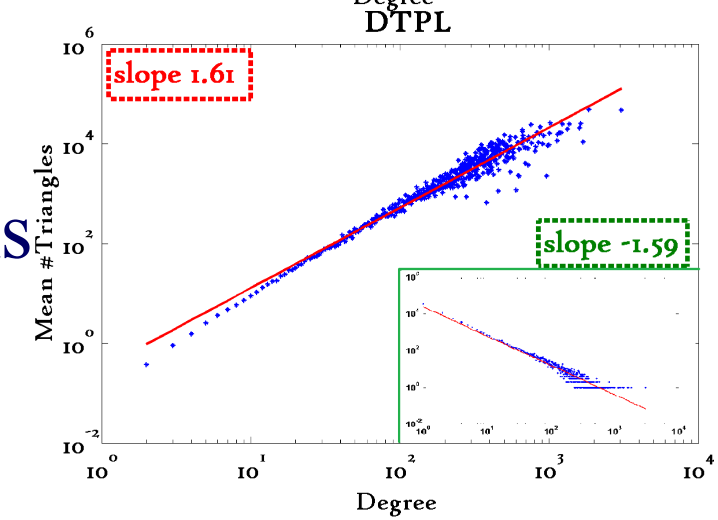
Reuters

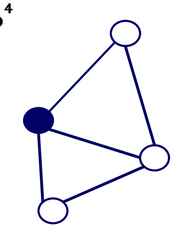


SN



Epinions



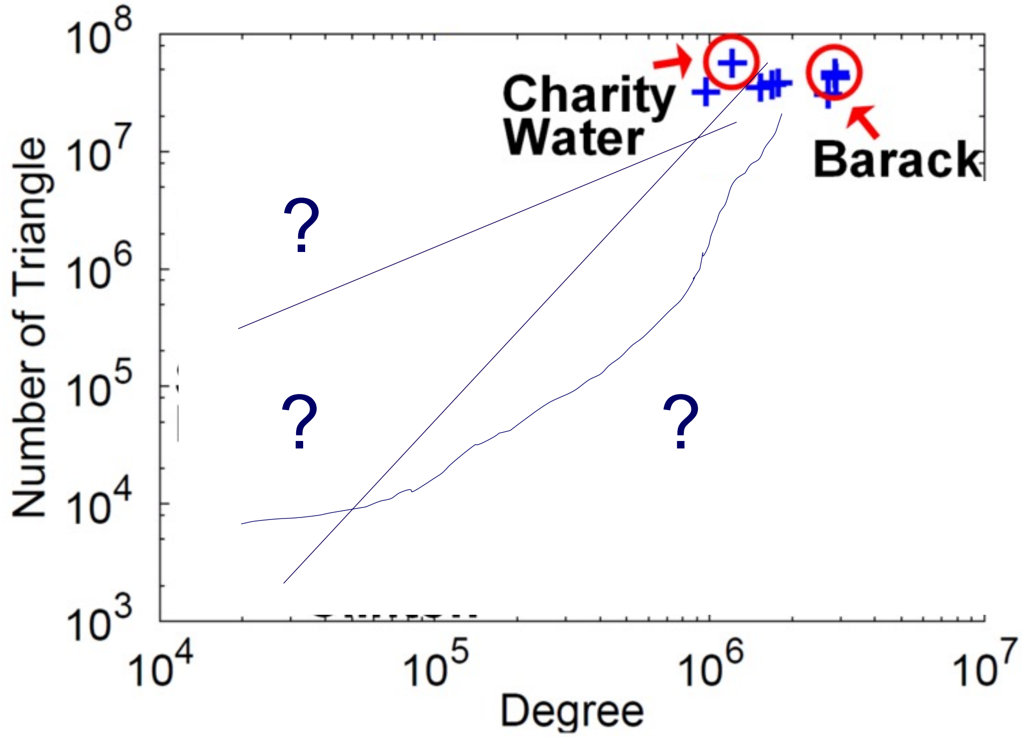


X-axis: degree

Y-axis: mean # triangles

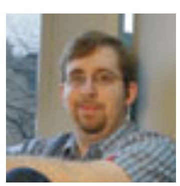
n friends -> ~$n^{1.6}$ triangles

Triangle counting for large graphs?

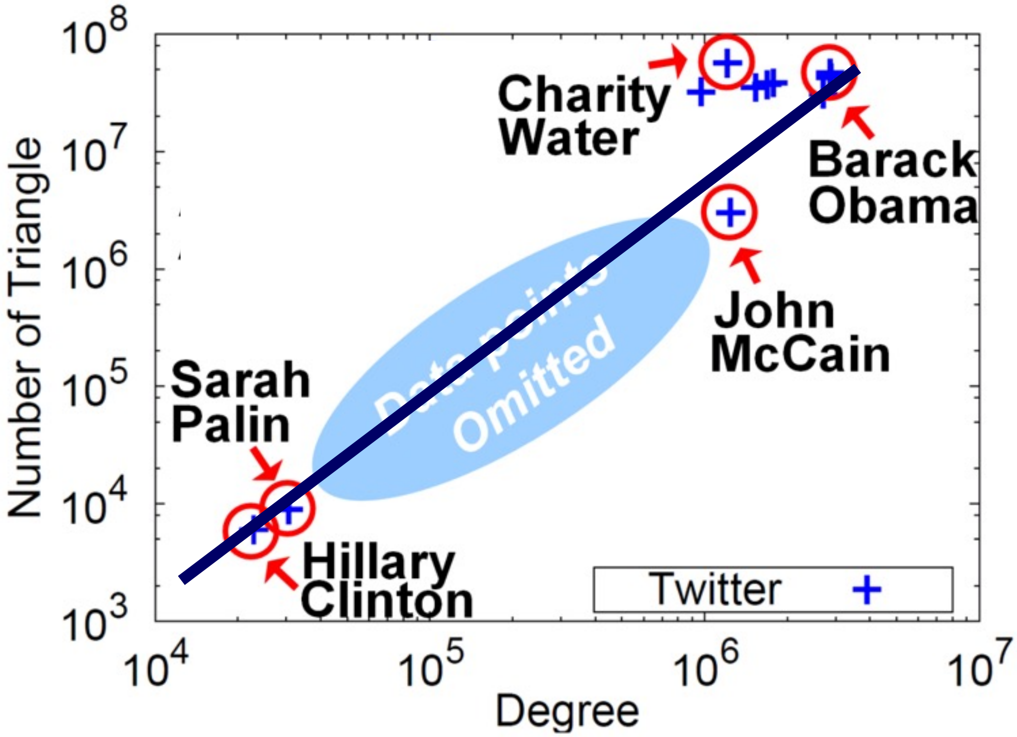


Anomalous nodes in Twitter(~ 3 billion edges)

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



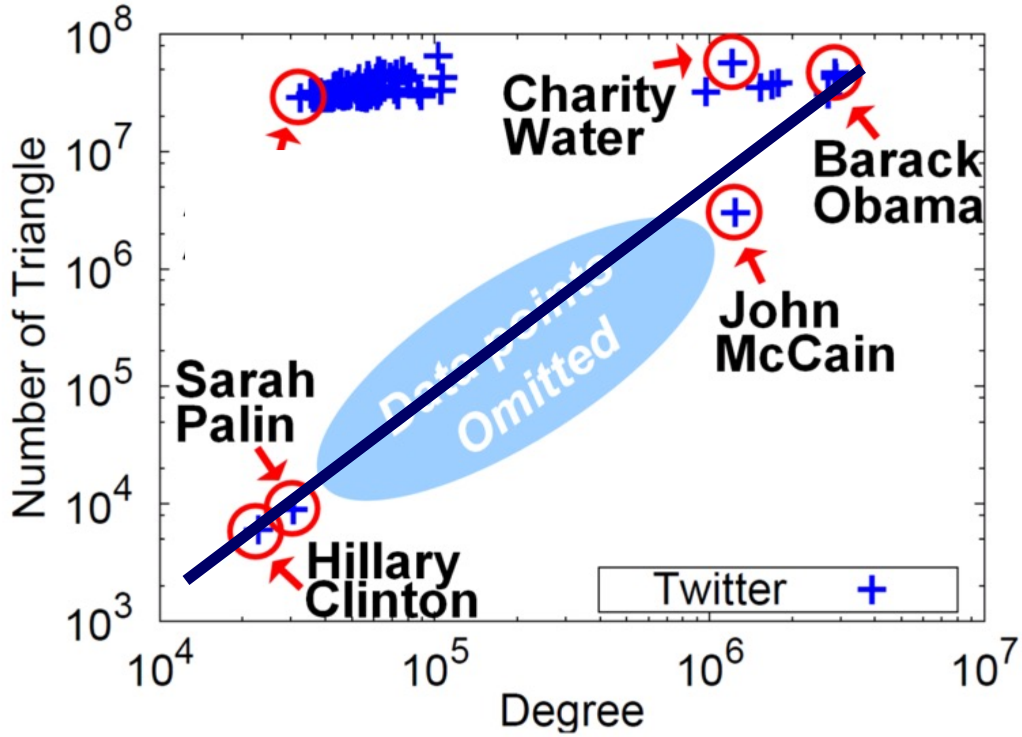
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

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

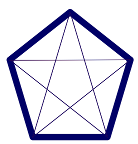
Triangle counting for large graphs?

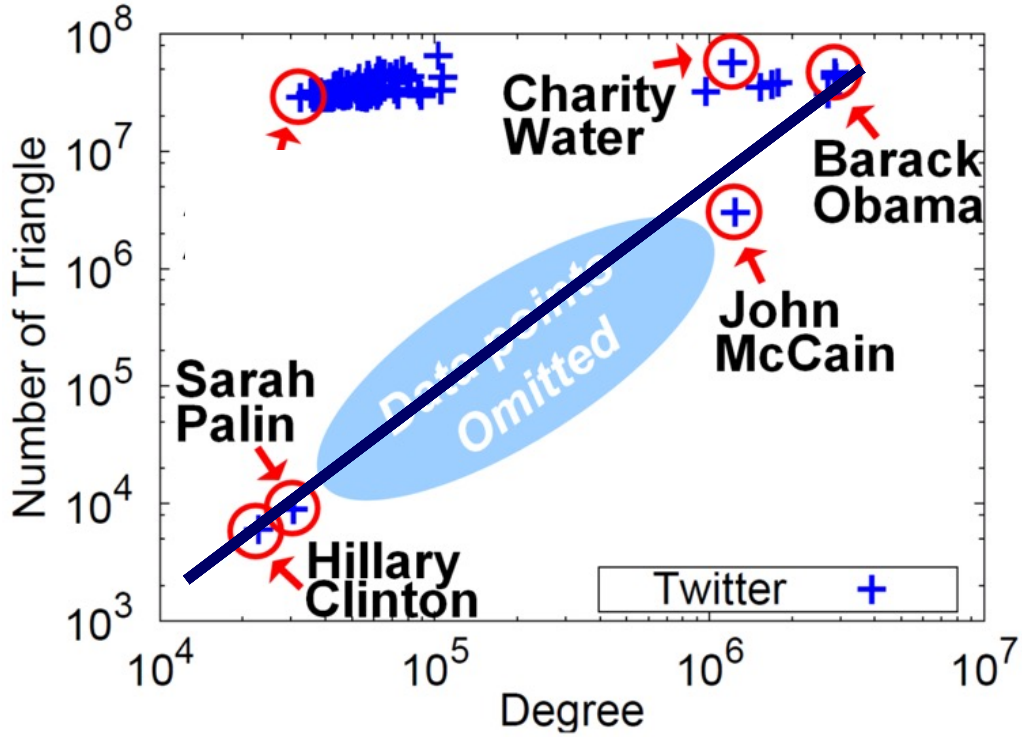


Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]

Triangle counting for large graphs?



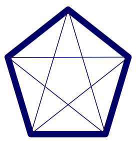


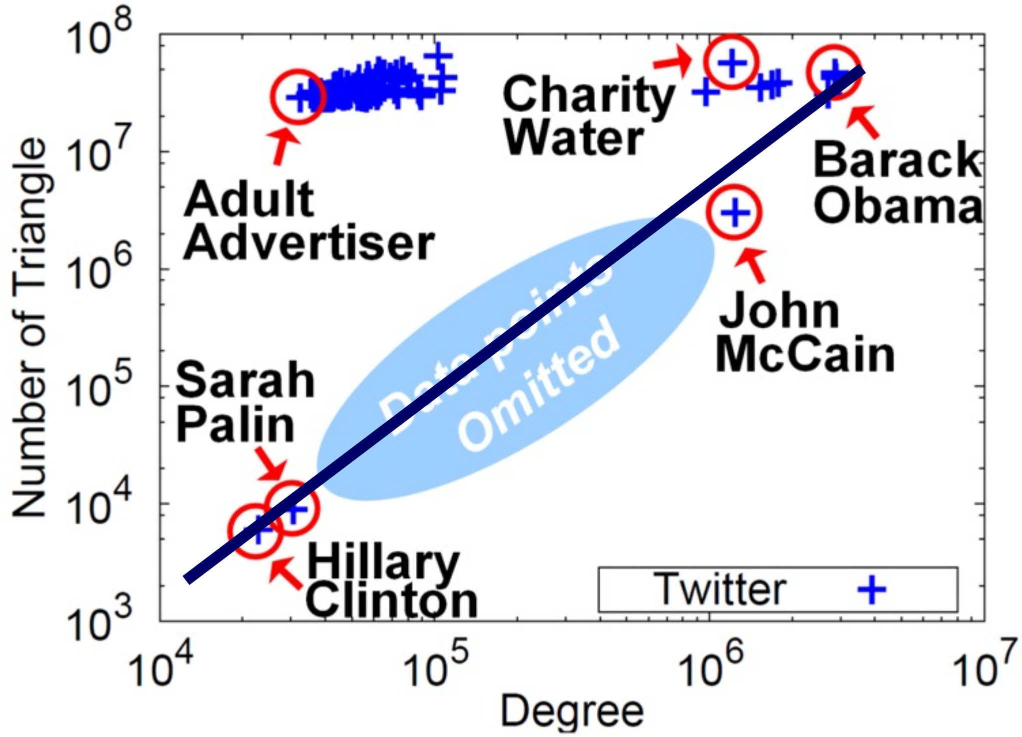
Anomalous nodes in Twitter(~ 3 billion edges)

[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]

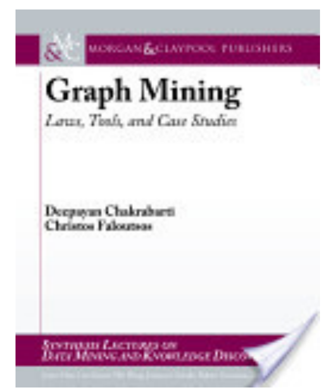
RTG: A Recursive Realistic Graph Generator using Random Typing Leman Akoglu and Christos Faloutsos. *PKDD*'09.

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.

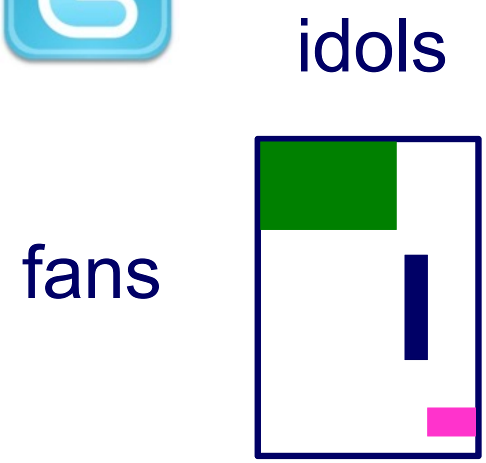
Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

Patterns  anomalies

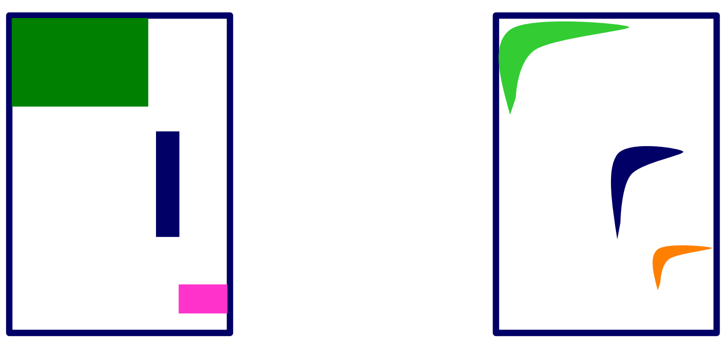
How to find 'suspicious' groups?

- 'blocks' are normal, right?



Except that:

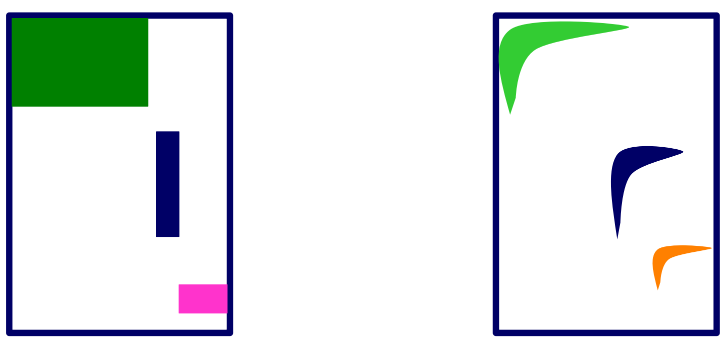
- ‘blocks’ are normal, ~~right~~?
- ‘hyperbolic’ communities are more realistic [Araujo+, PKDD’14]



Except that:

- ‘blocks’ are usually suspicious
- ‘hyperbolic’ communities are more realistic [Araujo+, PKDD’14]

Q: Can we spot blocks, easily?

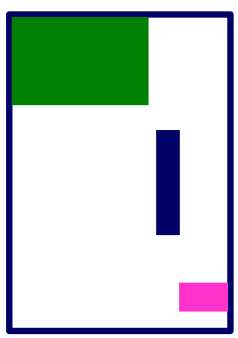


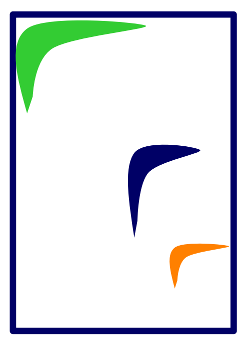
Except that:

- ‘blocks’ are usually suspicious
- ‘hyperbolic’ communities are more realistic [Araujo+, PKDD’14]

Q: Can we spot blocks, easily?

A: Silver bullet: SVD!

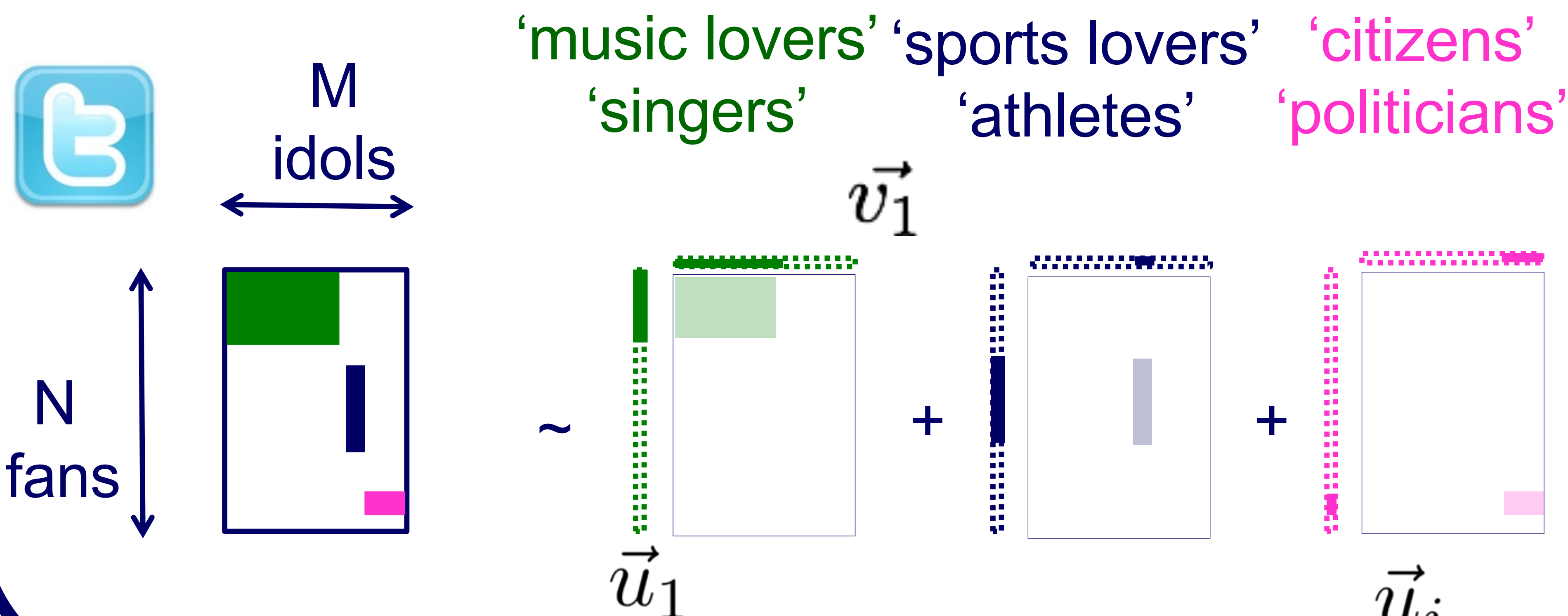




DETAILS

Crush intro to SVD

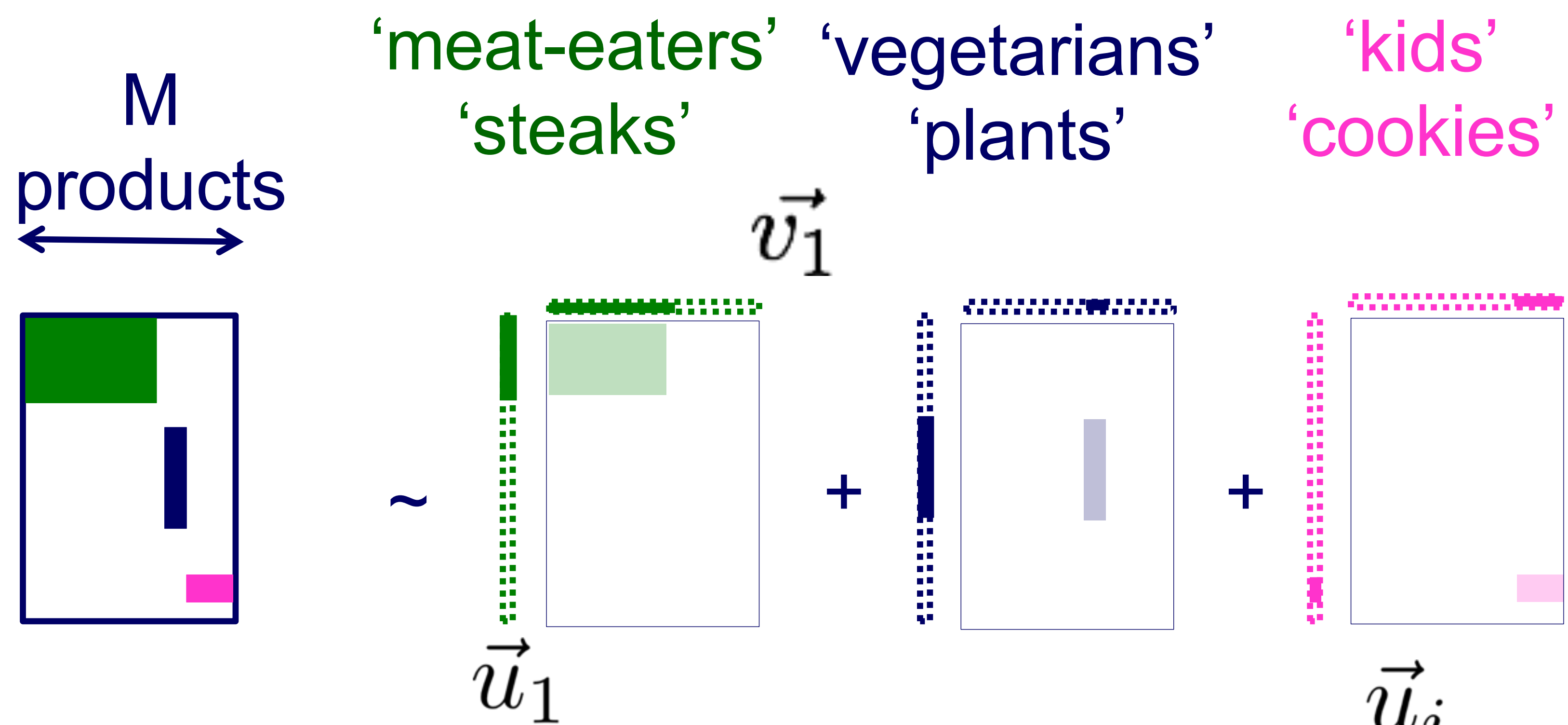
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

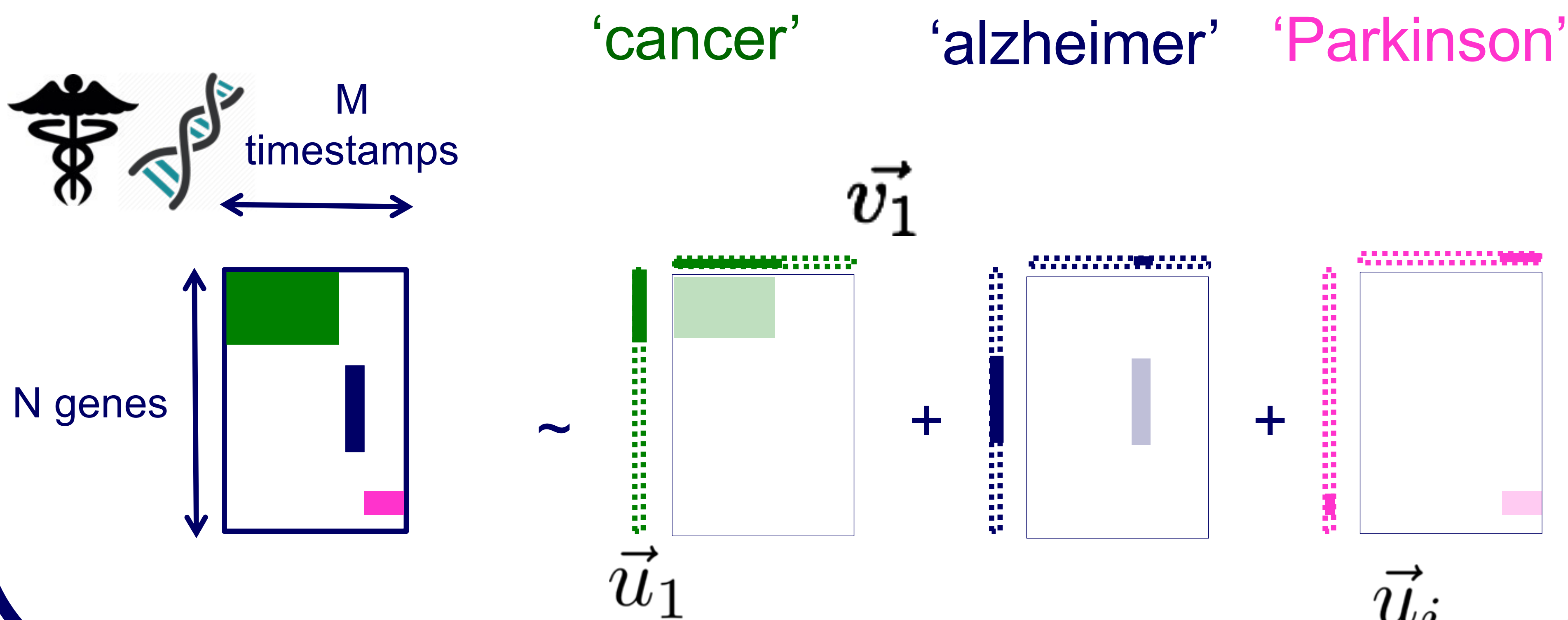
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

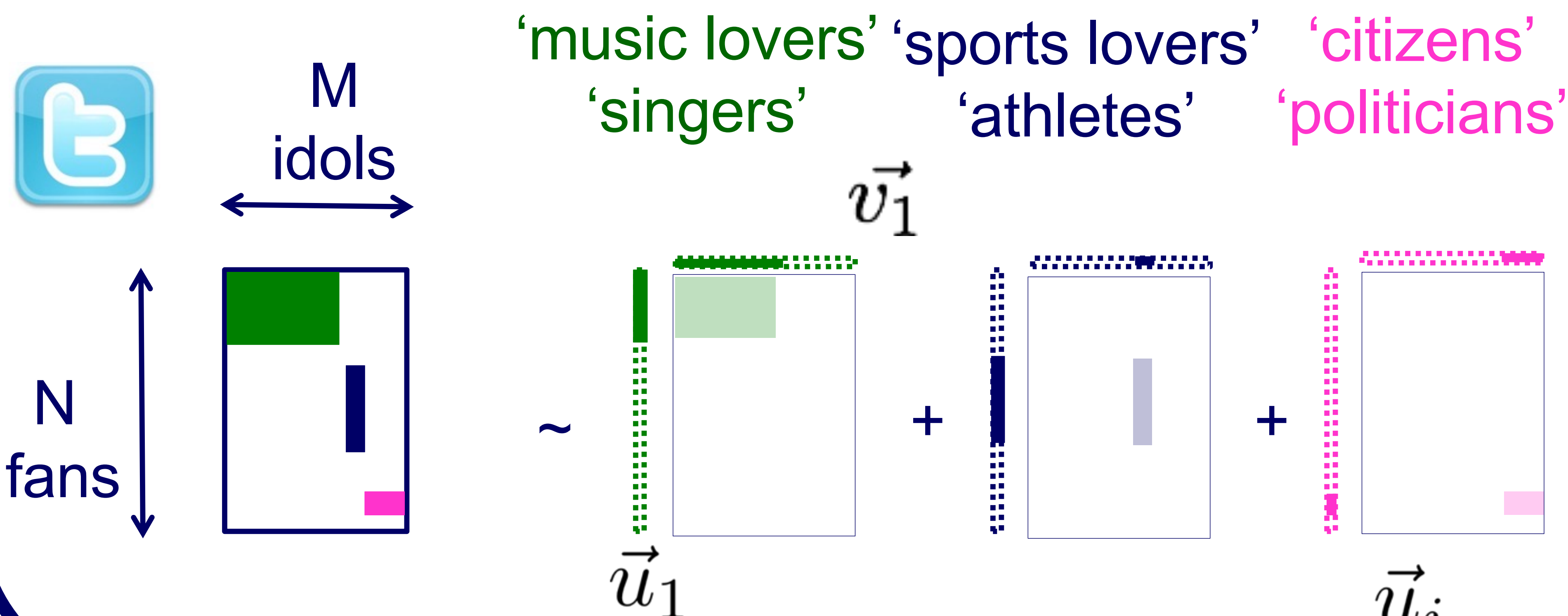
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

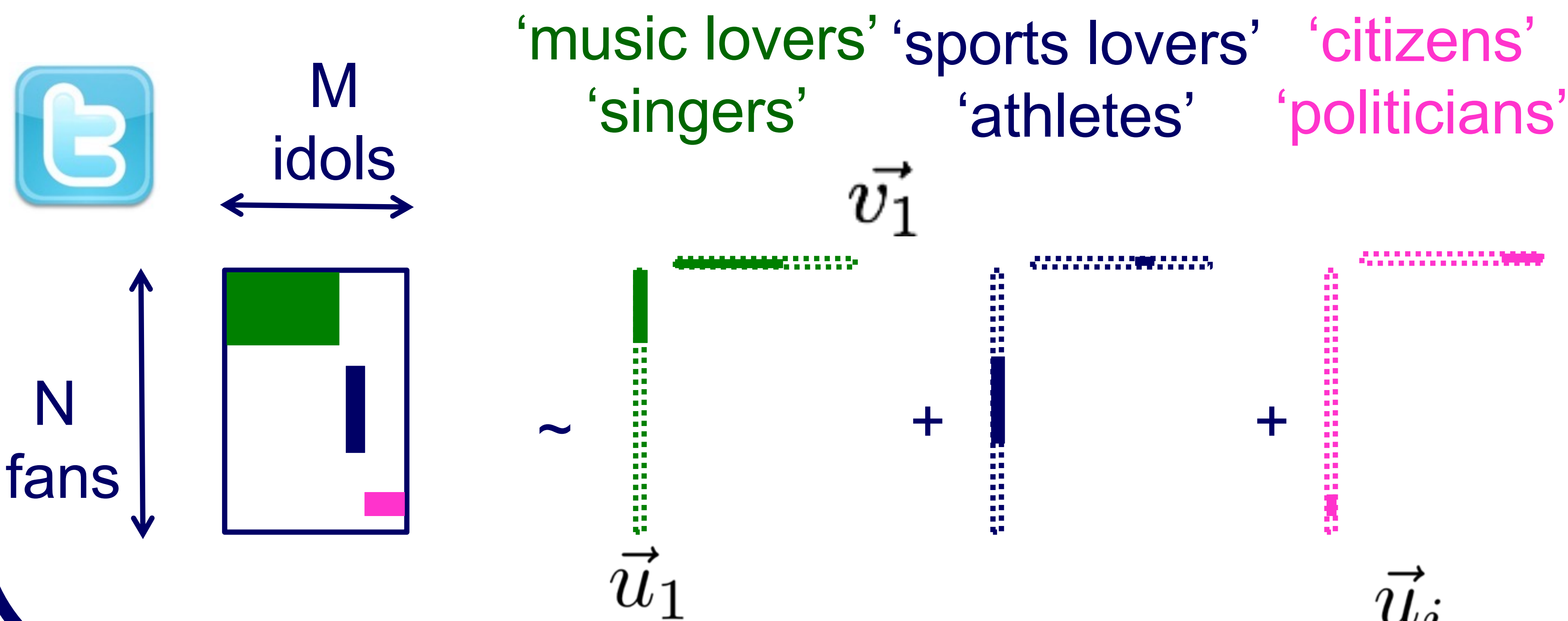
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

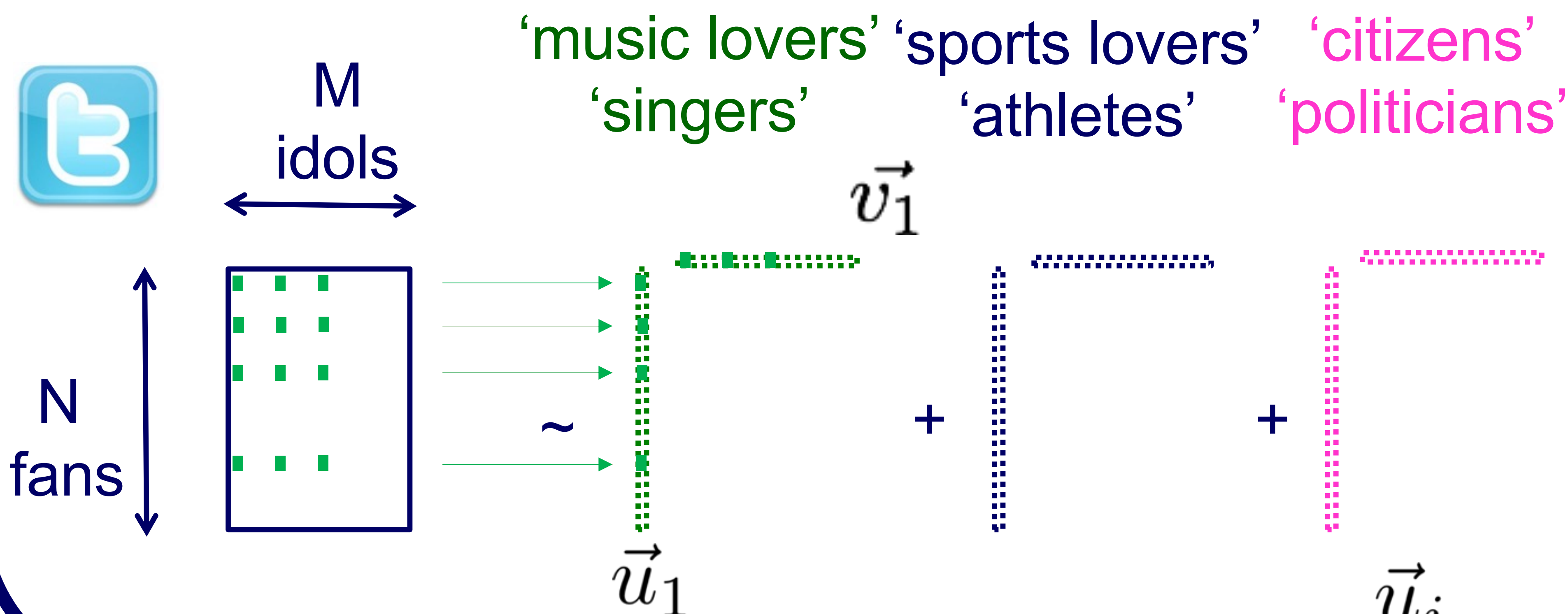
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks **Even if shuffled!**



Inferring Strange Behavior from Connectivity Pattern in Social Networks

PAKDD’14

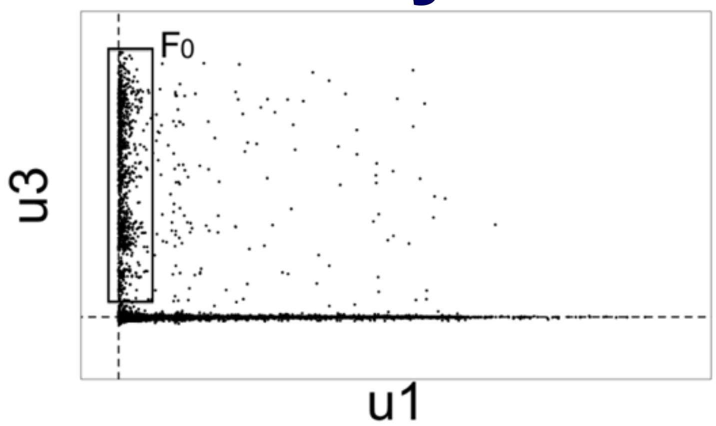
Meng Jiang, Peng Cui, Shiqiang Yang (Tsinghua)
Alex Beutel, Christos Faloutsos (CMU)

Dataset

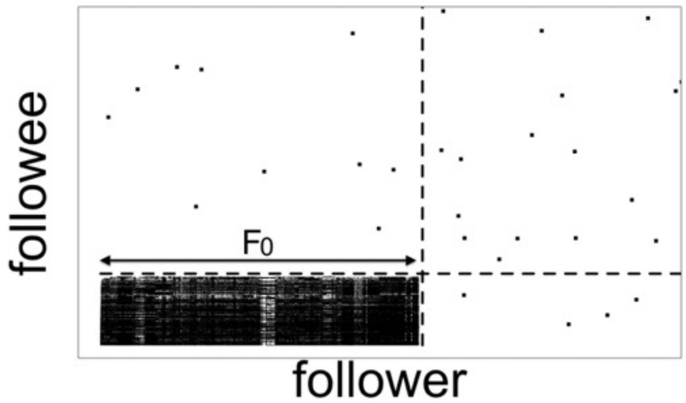
- Tencent Weibo
- 117 million nodes (with profile and UGC data)
- 3.33 billion directed edges

Real Data

"Rays"

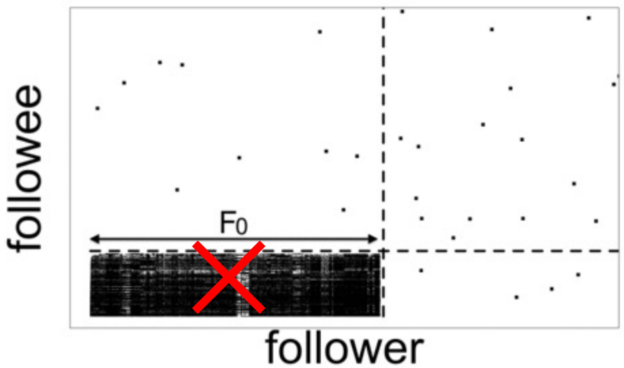


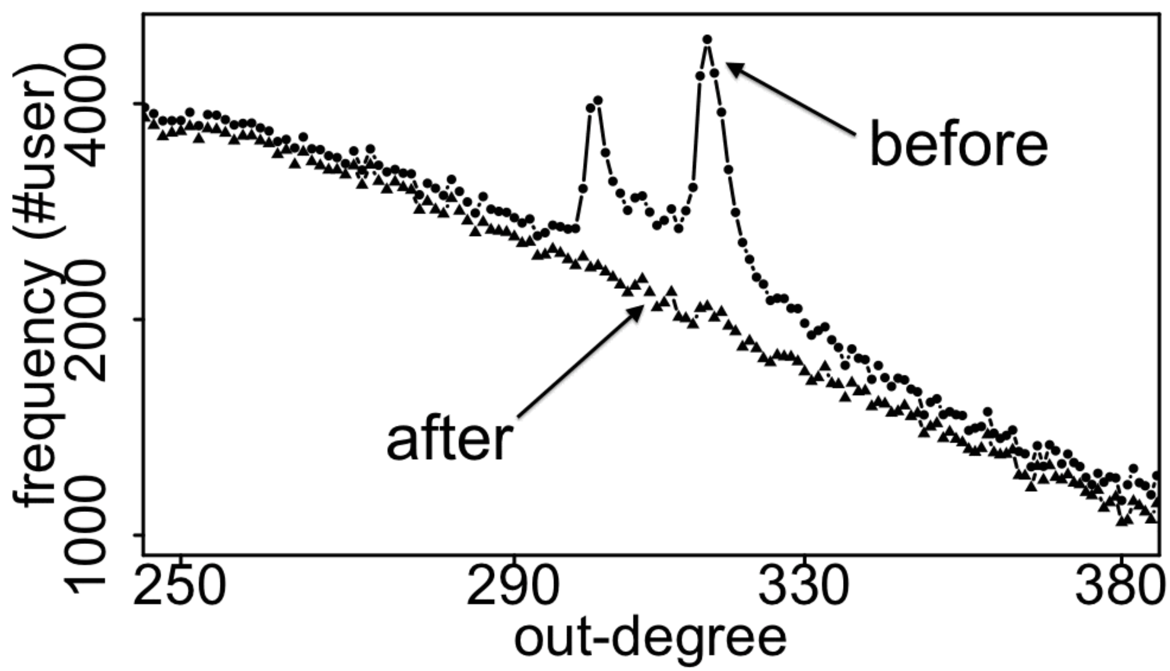
"Block"



Real Data

- Spikes on the out-degree distribution





Carnegie Mellon

Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral methods
    - No labels – accounting application
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

Carnegie Mellon

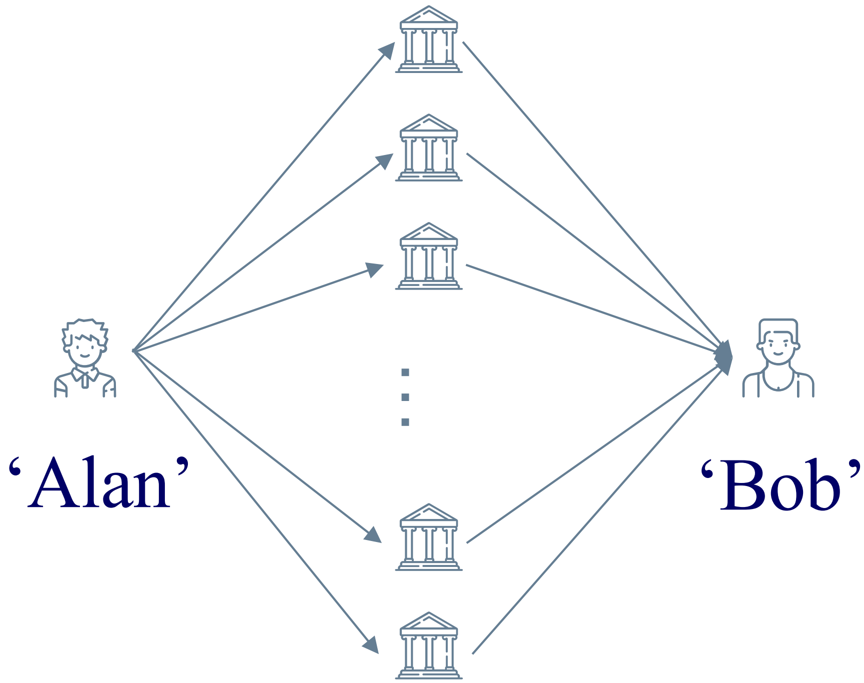
AutoAudit: Mining Accounting and Time-Evolving Graphs

IEEE Big Data, 2020

Meng-Chieh Lee[1], Yue Zhao[2], Aluna Wang[2], Pierre Jinghong Liang[2],
Leman Akoglu[2], Vincent S. Tseng[1], Christos Faloutsos[2]

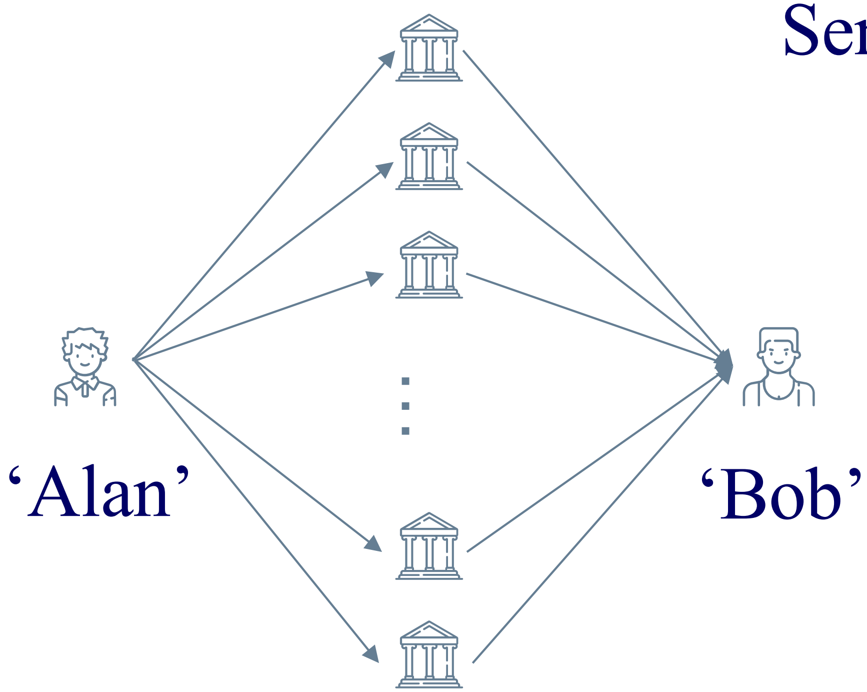
Carnegie Mellon University

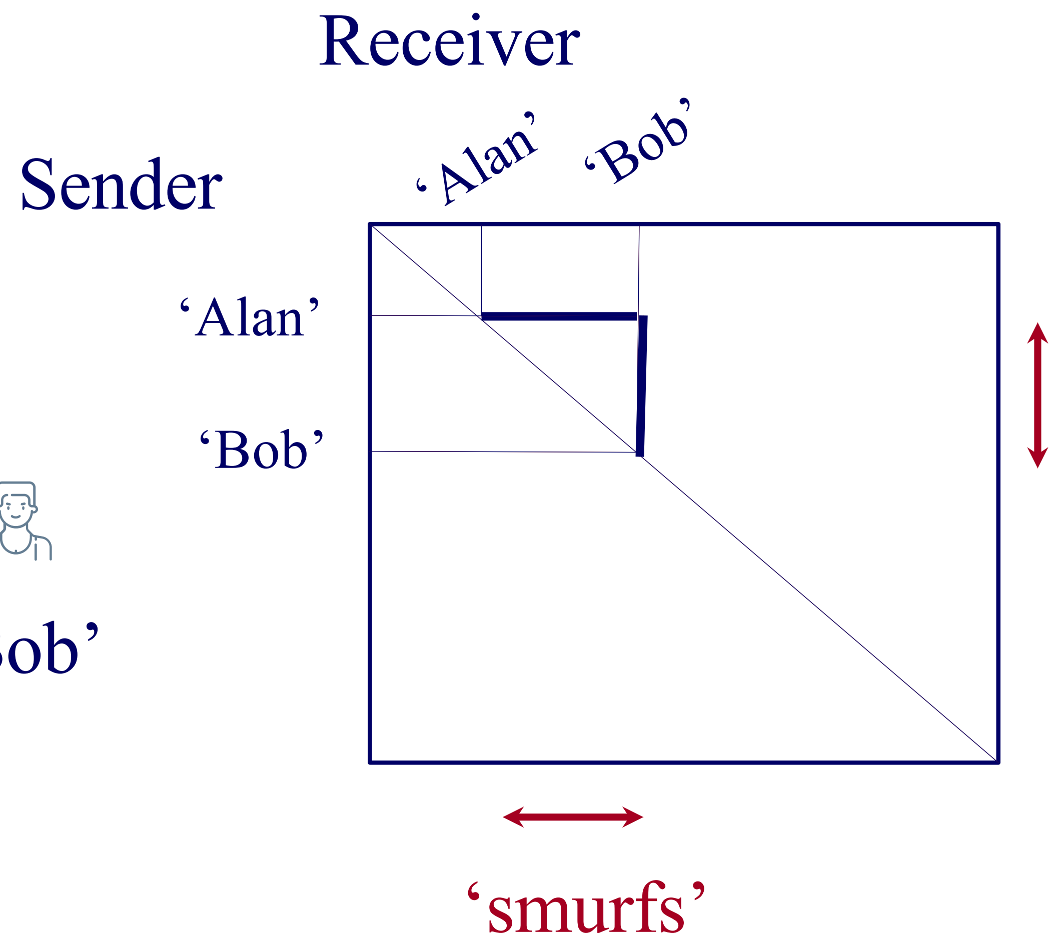
‘Smurfing’



How to spot it?

'Smurfing'

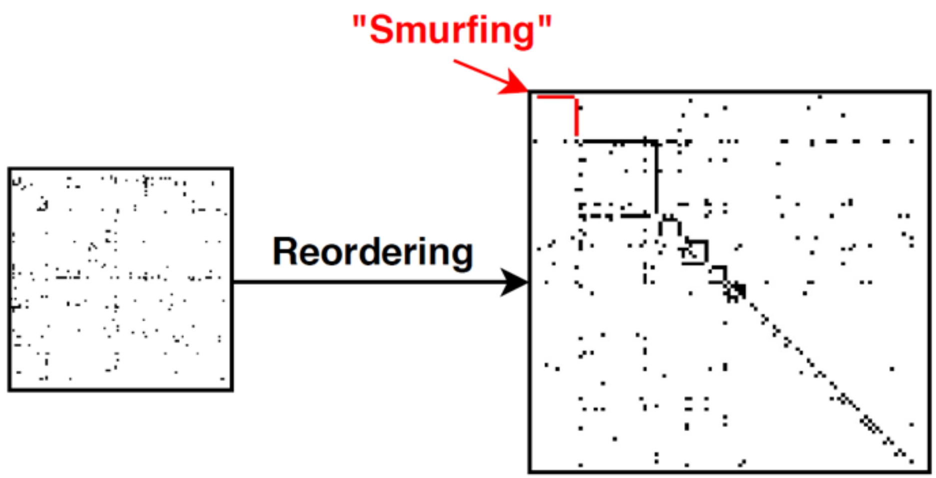




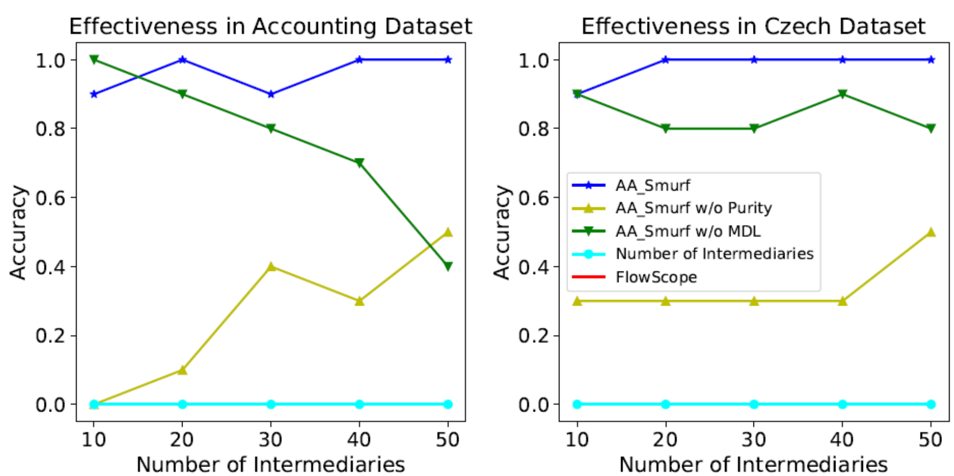
Reverse-'L' shape
(after careful re-ordering)

Carnegie Mellon

AutoAudit: Experiments



accuracy



← Ideal: 100%

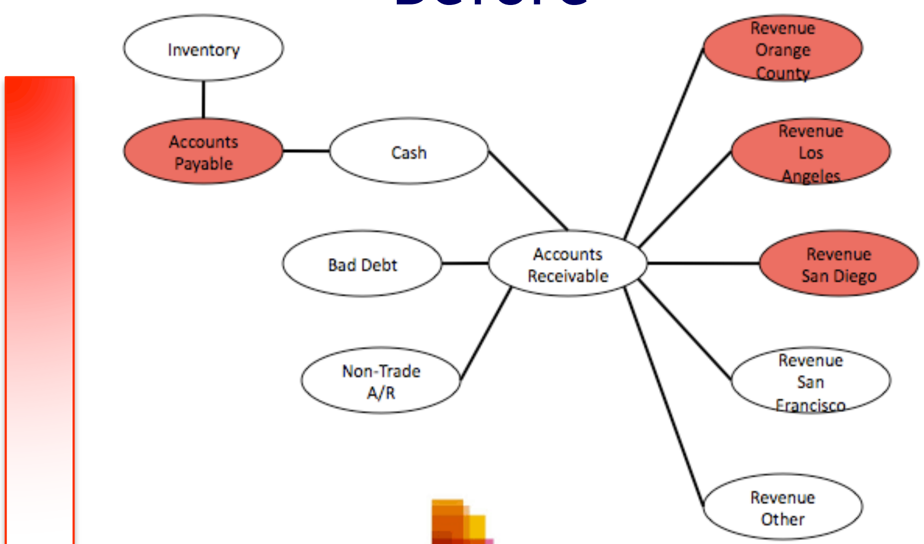
Roadmap

- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral methods
    - No labels – dense subgraphs
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

Network Effect Tools: SNARE

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

Before

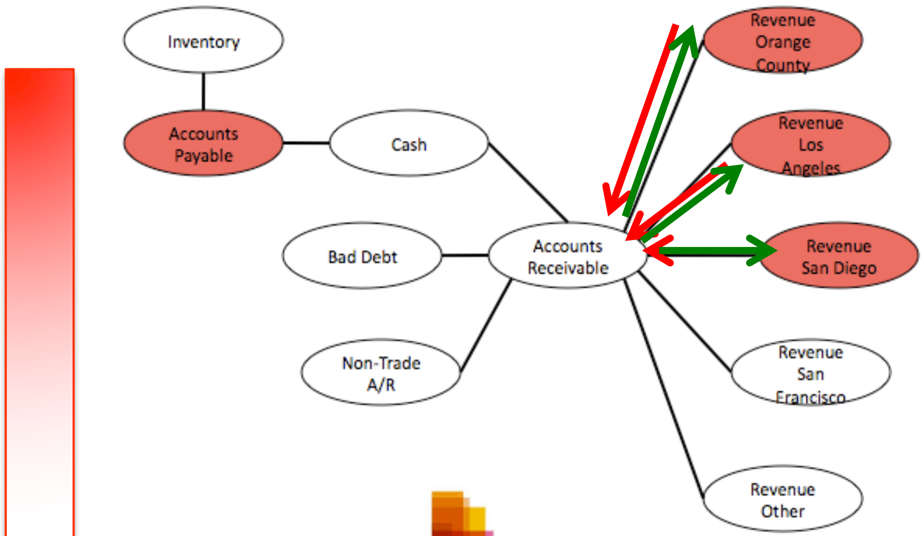


pwc

Network Effect Tools: SNARE

- A: Belief Propagation.

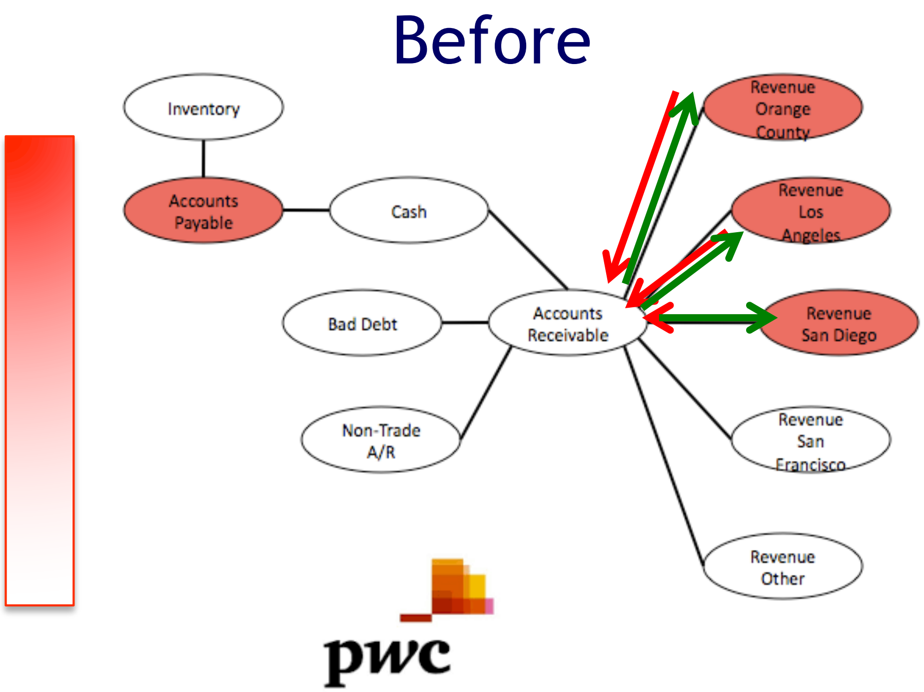
Before

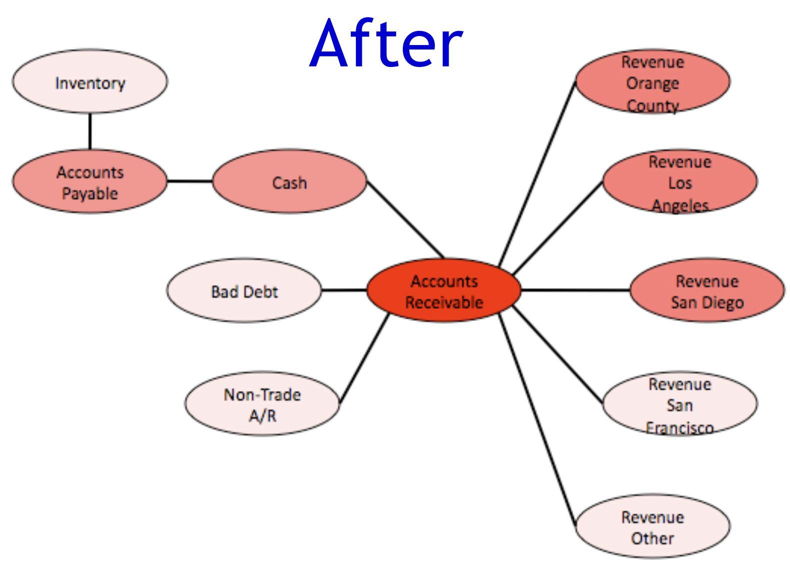


Carnegie Mellon

Network Effect Tools: SNARE

- A: Belief Propagation.



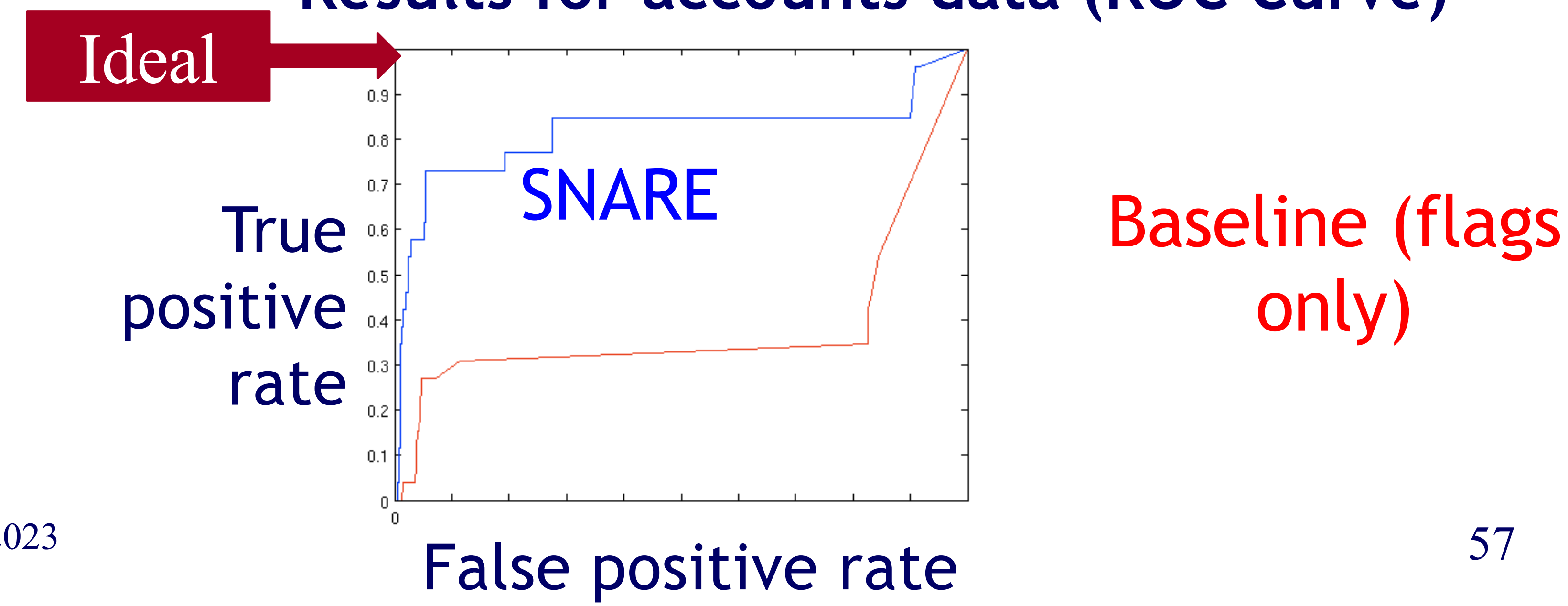


Mary McGlohon, Stephen Bay, Markus G. Anderle, David M. Steier, Christos Faloutsos: *SNARE: a link analytic system for graph labeling and risk detection*. KDD 2009: 1265-1274

Network Effect Tools: SNARE

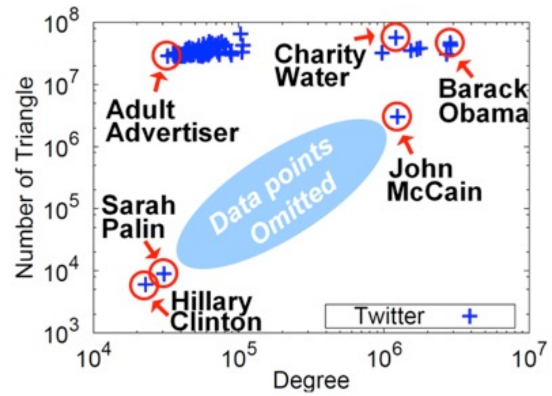
- Produces improvement over simply using flags
  - Up to 6.5 lift
  - Improvement especially for low false positive rate

Results for accounts data (ROC Curve)



Summary of Part#1

- *many* patterns in real graphs
  - Power-laws everywhere
  - Long (and growing) list of tools for anomaly/fraud detection



Patterns  anomalies

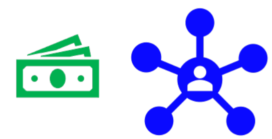
Roadmap

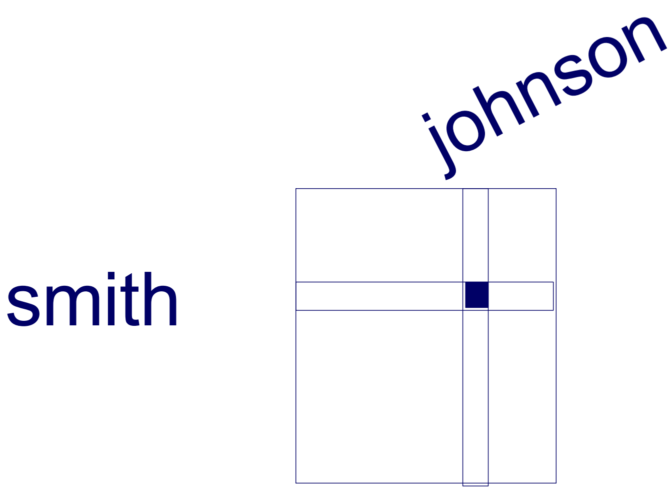
- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs
  - P2.1: tools/tensors
  - P2.2: other patterns
- Conclusions

Part 2: Time evolving graphs; tensors

Graphs over time -> tensors!

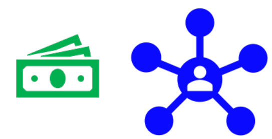
- Problem #2.1:
  - Given who calls whom, and when
  - Find patterns / anomalies

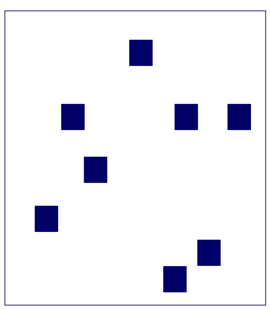




Graphs over time -> tensors!

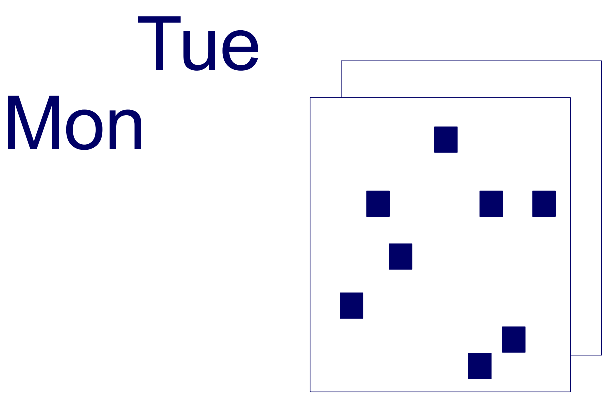
- Problem #2.1:
  – Given who calls whom, and when
  – Find patterns / anomalies





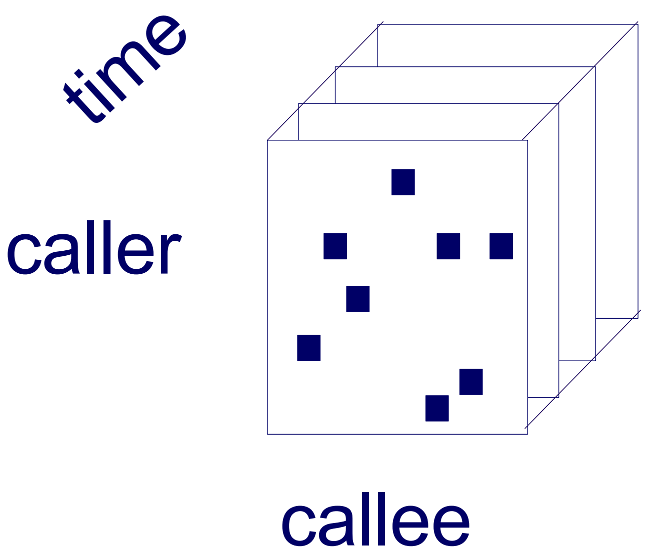
Graphs over time -> tensors!

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



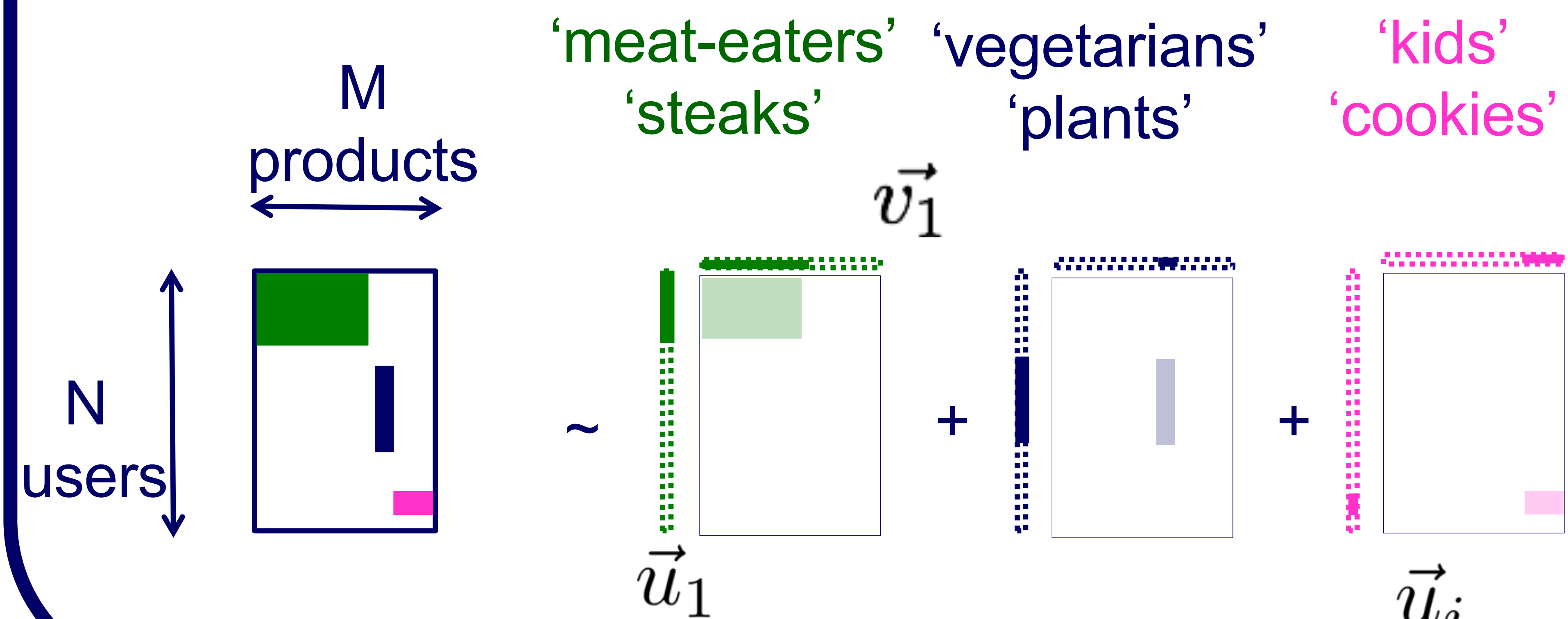
Graphs over time -> tensors!

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



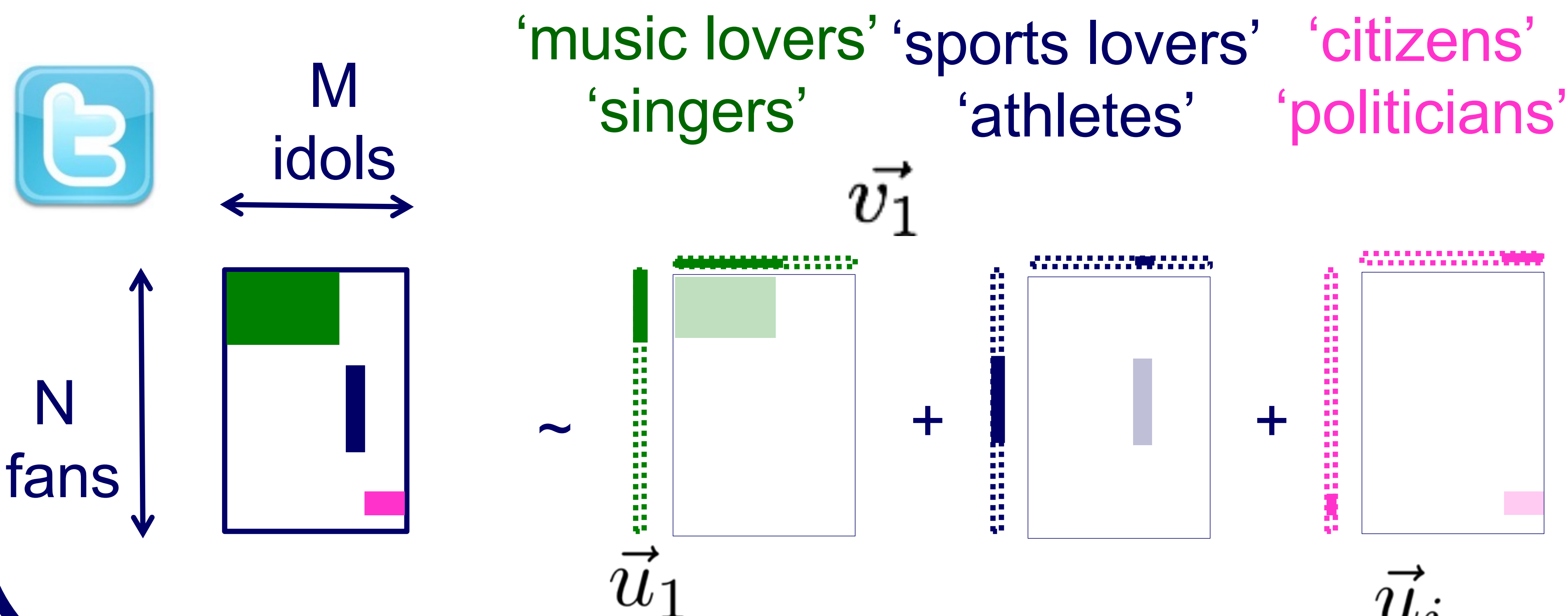
Answer : tensor factorization

- Recall: (SVD) matrix factorization: finds blocks



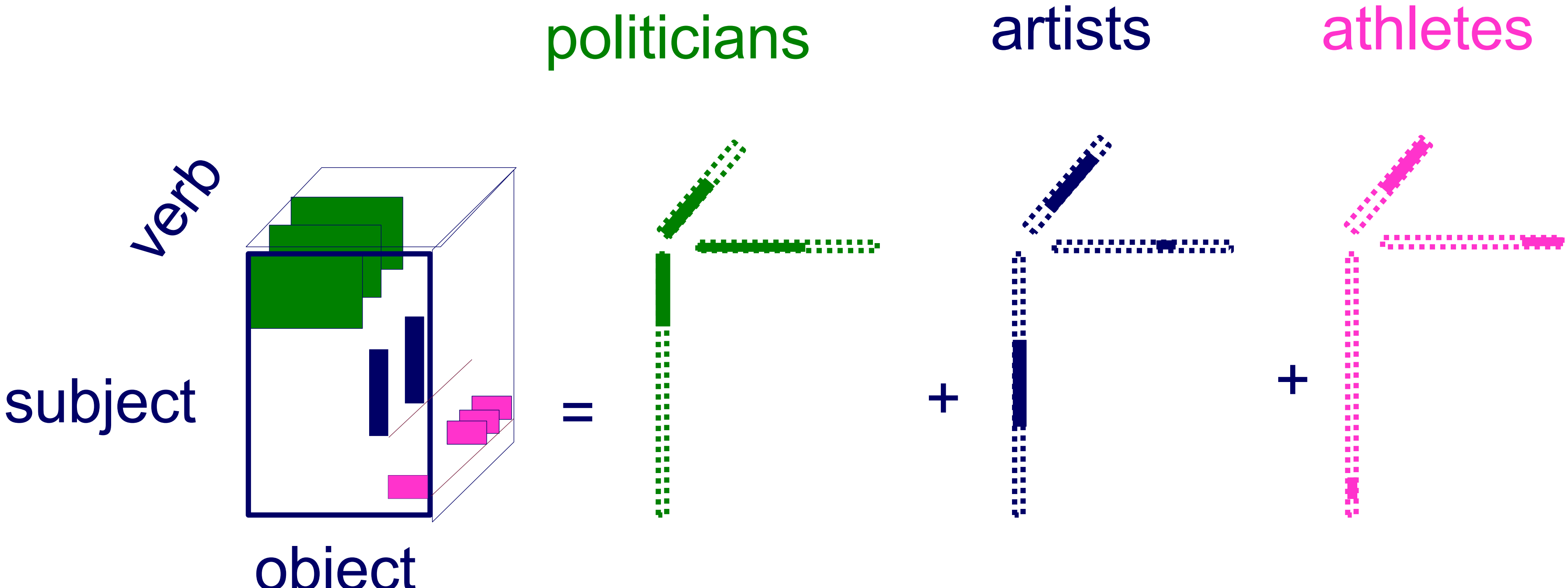
Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks



Answer: tensor factorization

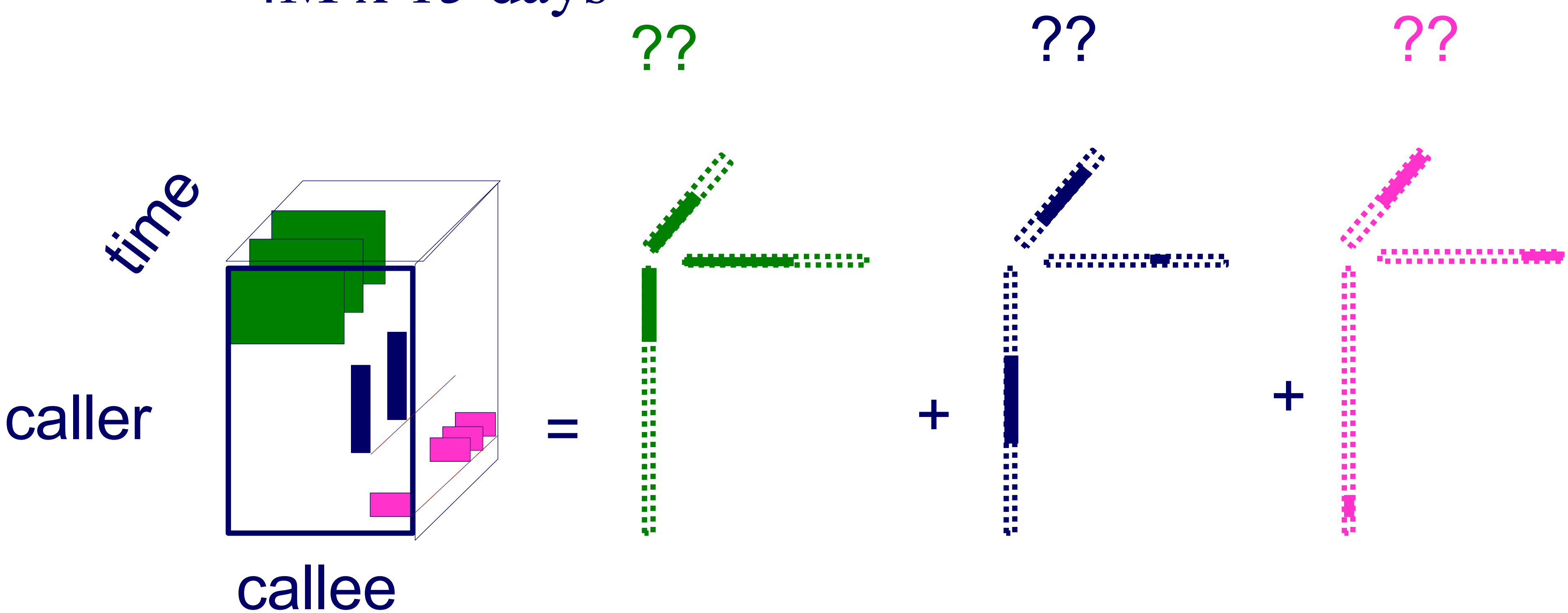
- PARAFAC decomposition



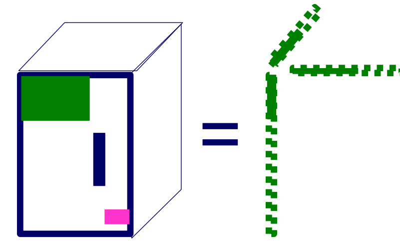
Carnegie Mellon

Answer: tensor factorization

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

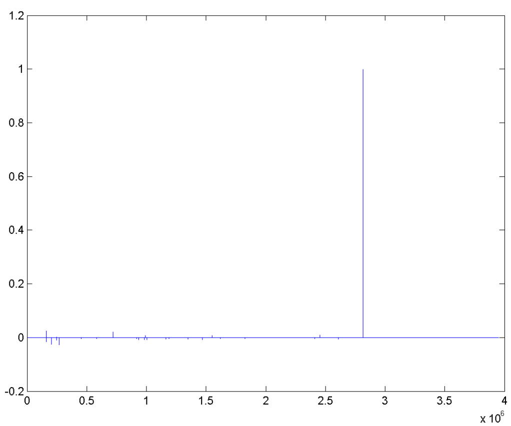


Anomaly detection in time-evolving graphs

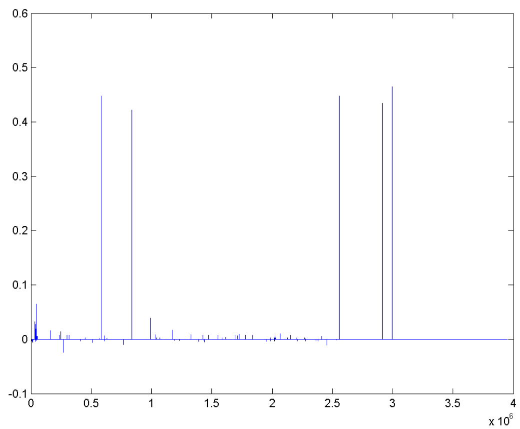


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

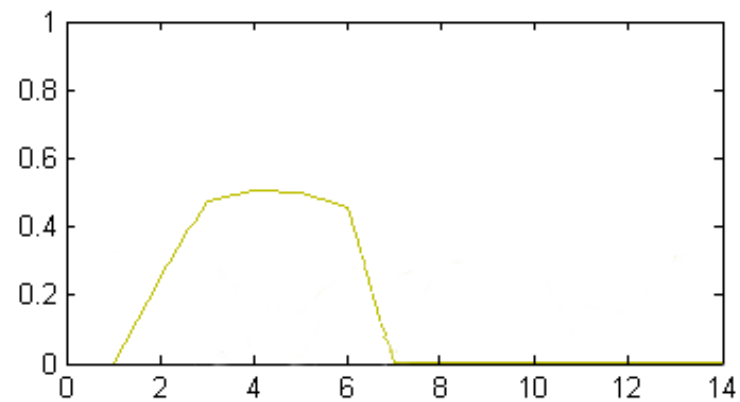
1 caller



5 receivers



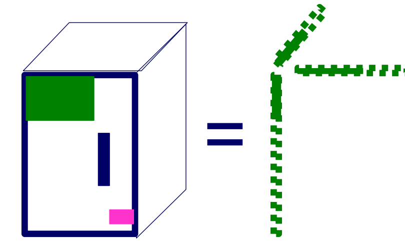
4 days of activity



~200 calls to EACH receiver on EACH day!

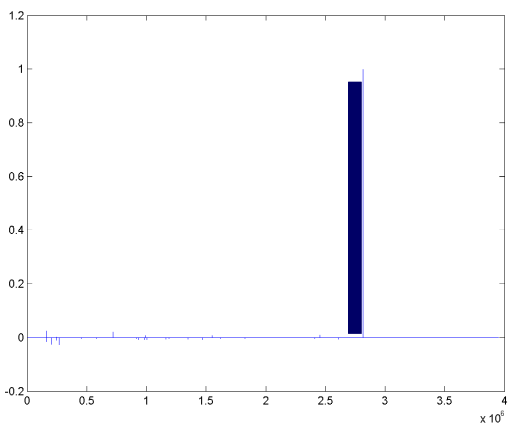
Carnegie Mellon

Anomaly detection in time-evolving graphs

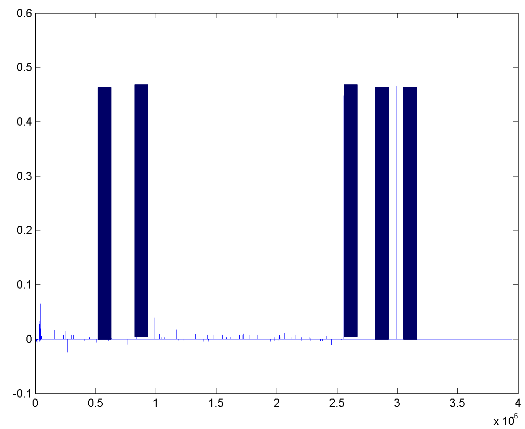


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

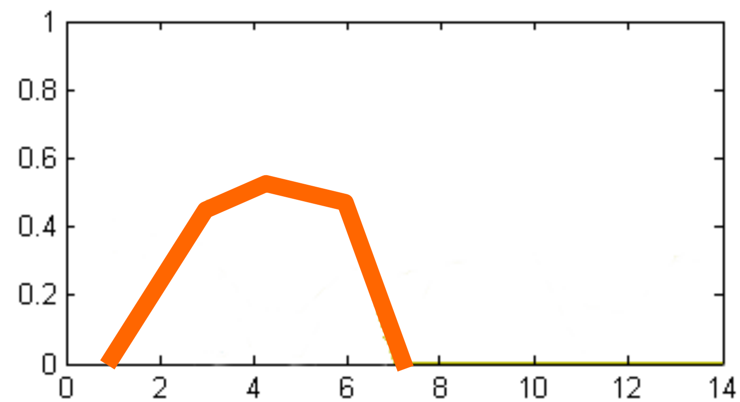
1 caller



5 receivers



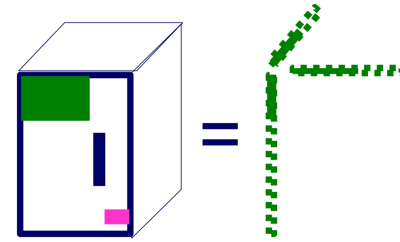
4 days of activity



~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs

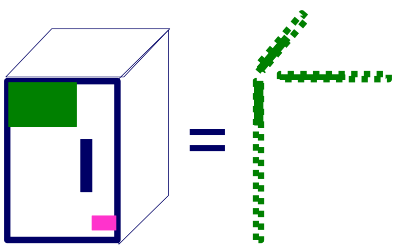


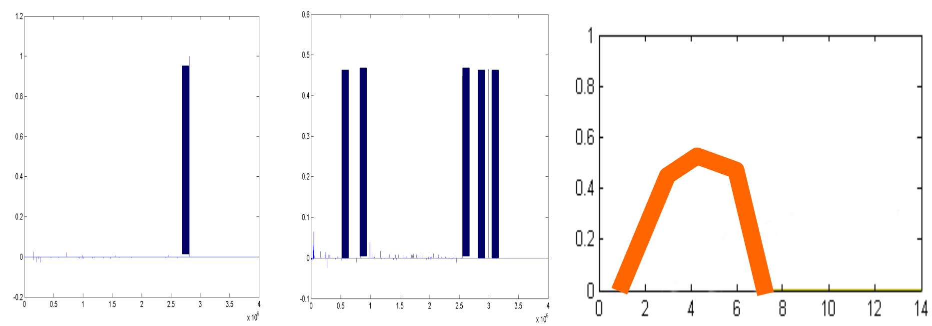
- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

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.

Part 2: Conclusions

- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- Surprising temporal patterns





Roadmap

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

Carnegie Mellon

TgraphSpot: Fast and Effective Anomaly Detection for Time-Evolving Graphs

IEEE BigData, 2022

Mirela Cazzolato[1,2], Saranya Vijayakumar[1], Xinyi Zheng[1], Namyong Park[1], Meng-Chieh Lee[1], Pedro Fidalgo[3,4], Bruno Lages[3], Agma J. M. Traina[2], Christos Faloutsos[1]

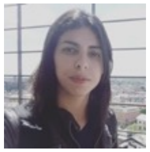
Carnegie Mellon USP iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

Open source:
https://github.com/mtcazzolato/tgraph-spot

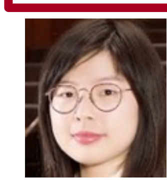
Video: https://youtu.be/jI1adN-BQuo?t=1537

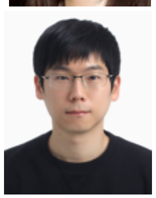
Carnegie Mellon

Authors


Mirela Cazzolato

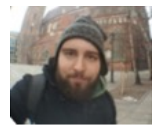

Saranya Vijayakumar


Xinyi Zheng


Namyong Park


Meng-Chieh Jeremy Lee

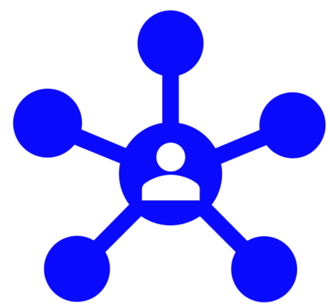

Pedro Fidalgo

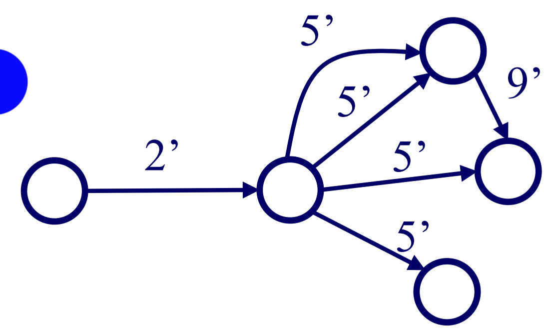

Bruno Lages


Agma Traina


Christos Faloutsos

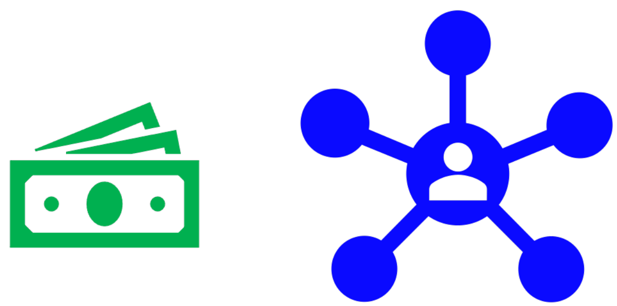
Problem definition

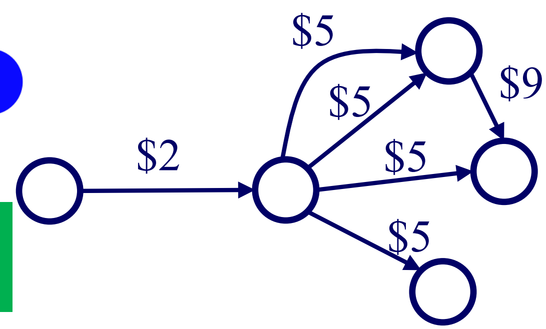




(source, destination, timestamp, duration)

Problem definition



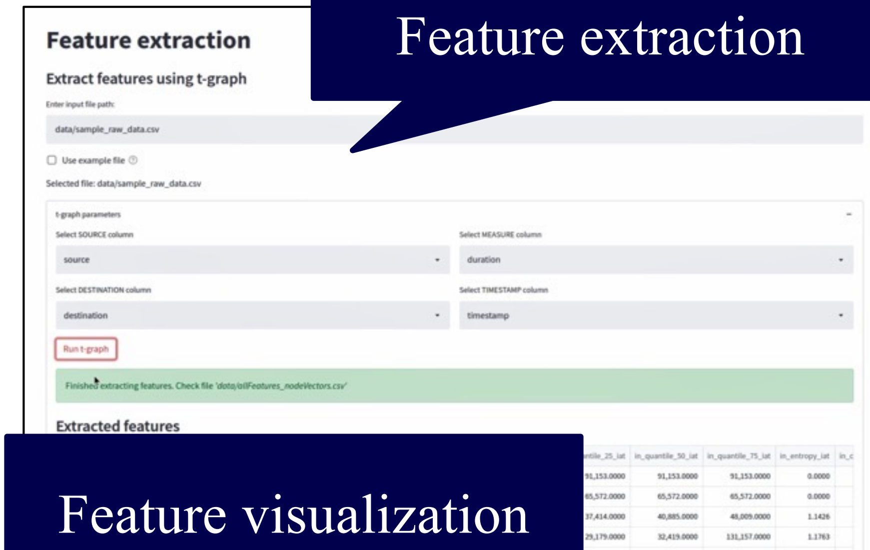


(source, destination, timestamp, $amount)

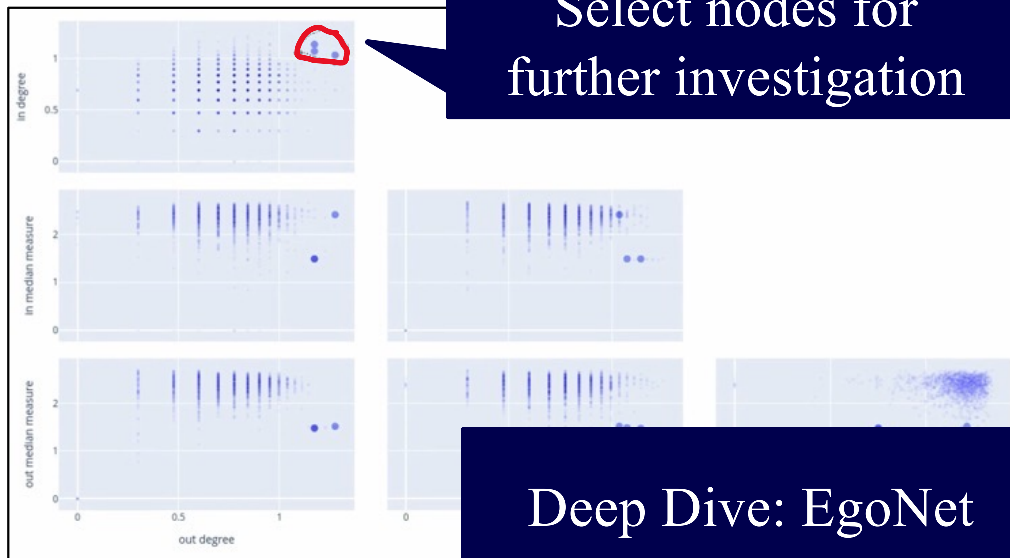
Carnegie Mellon

System Overview - current

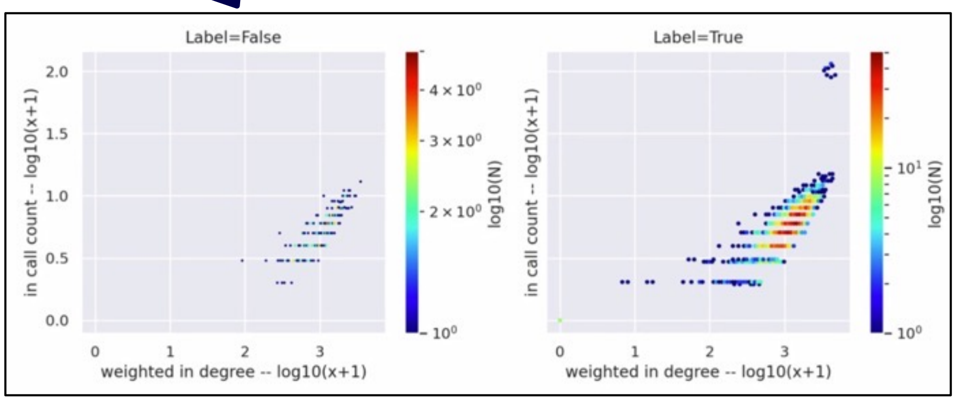
Feature extraction



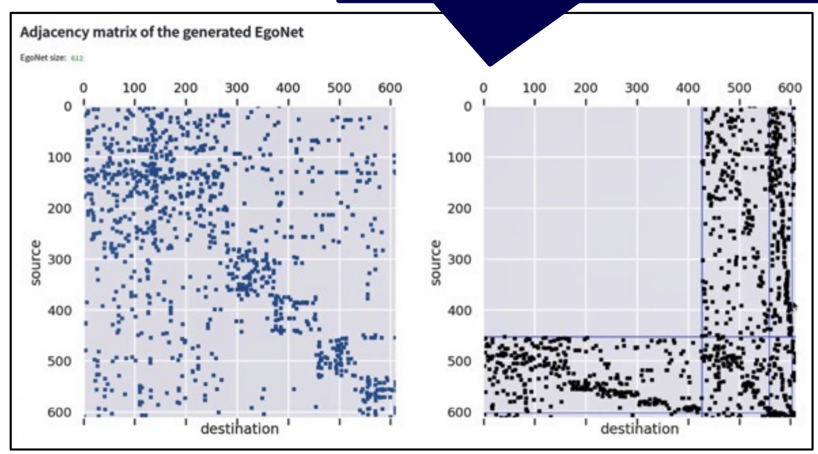
Select nodes for further investigation



Feature visualization



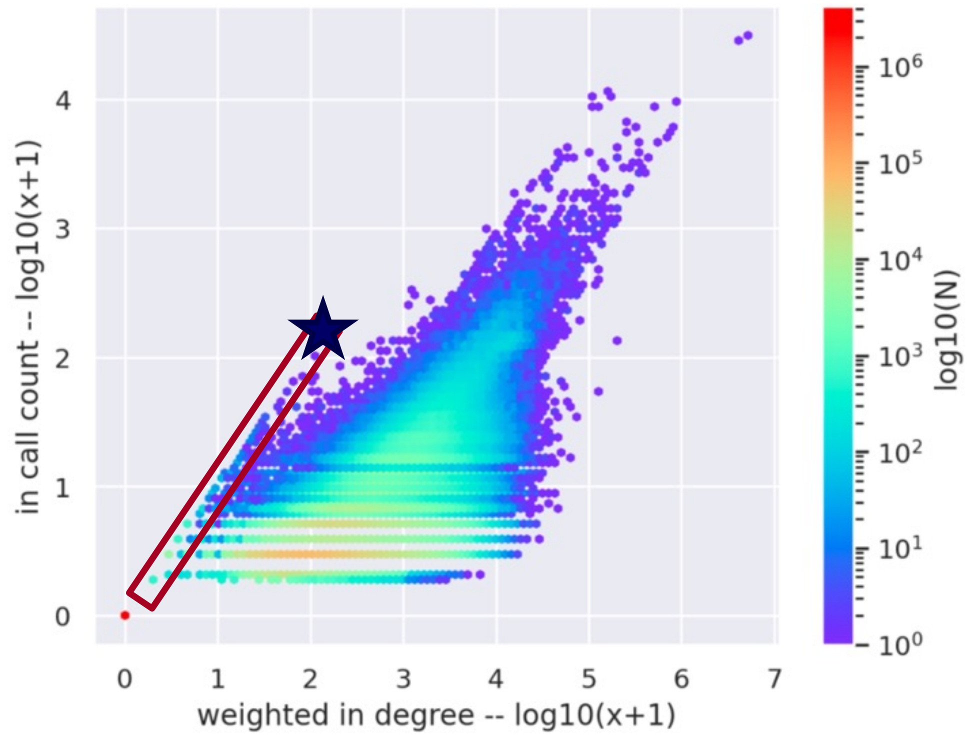
Deep Dive: EgoNet



Video: https://youtu.be/jI1adN-BQuo?t=1537

Discovery #1

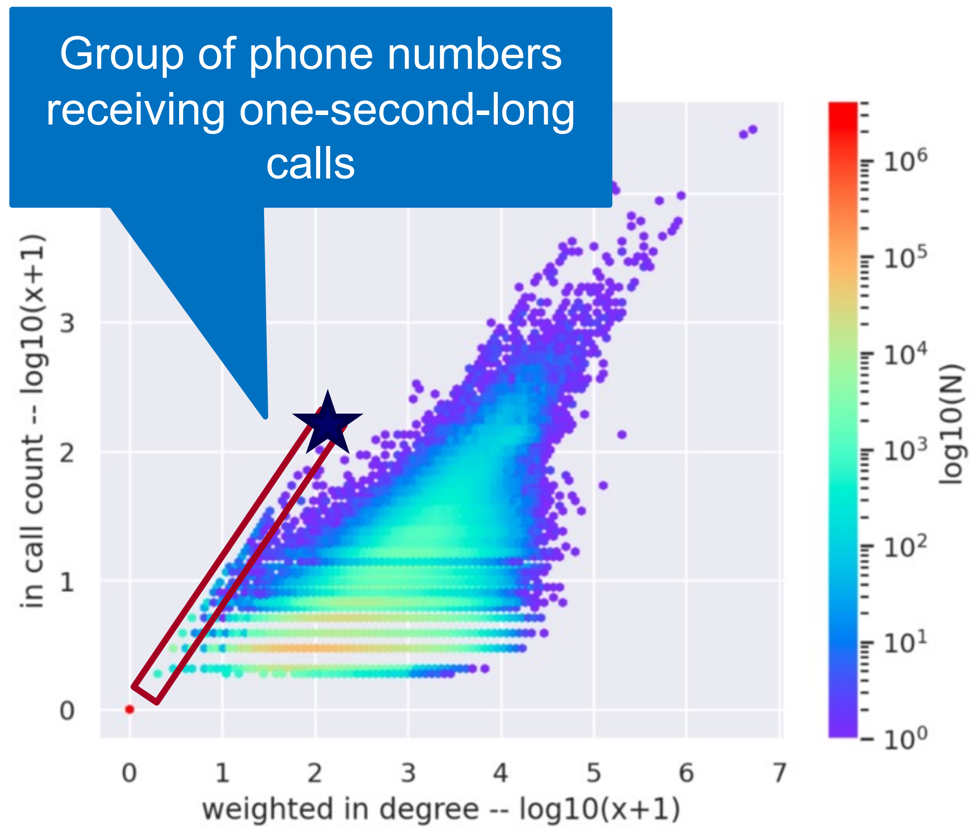
in-degree



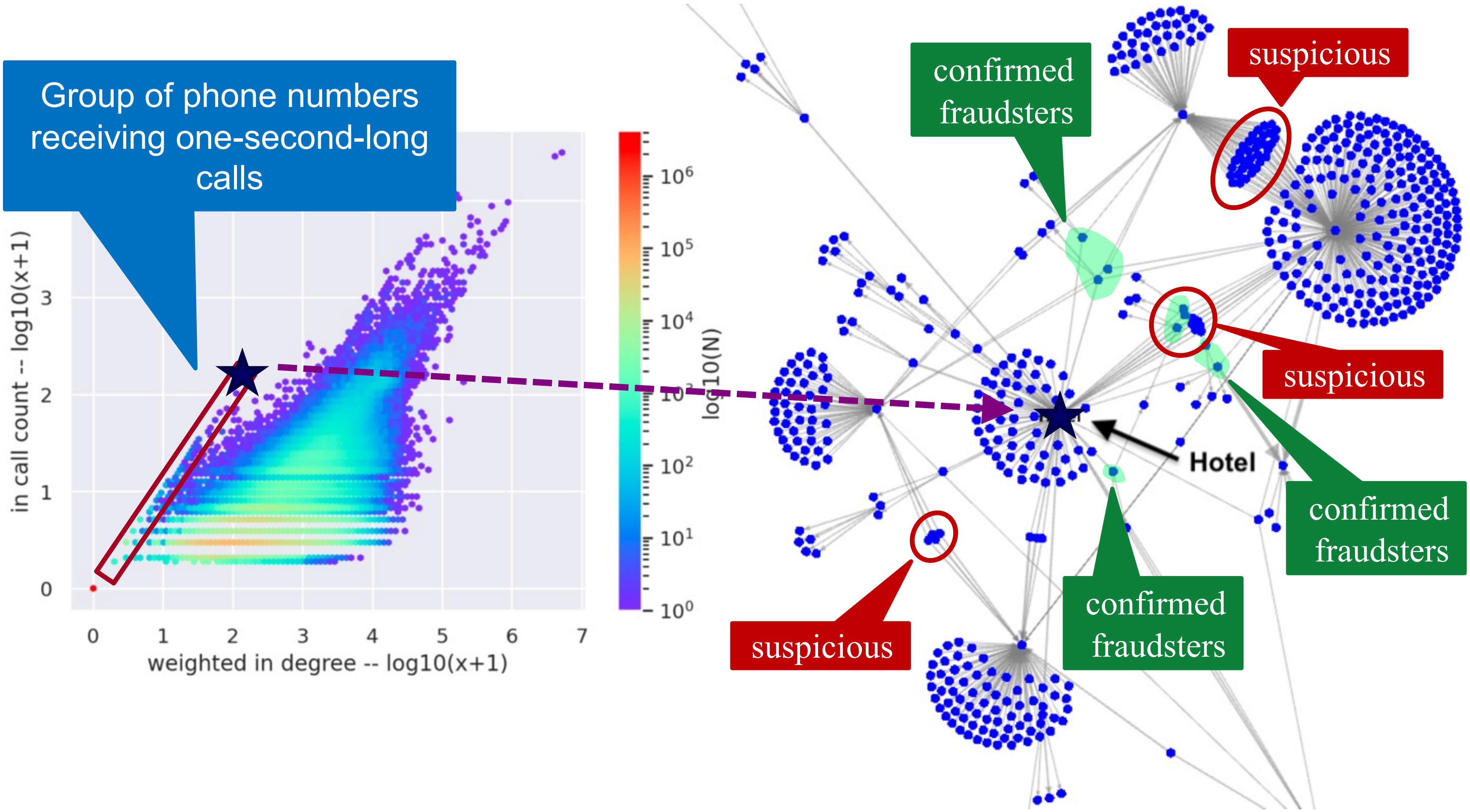
Weighted in-degree (= in-seconds)

Discovery #1

100 in-calls
100 seconds



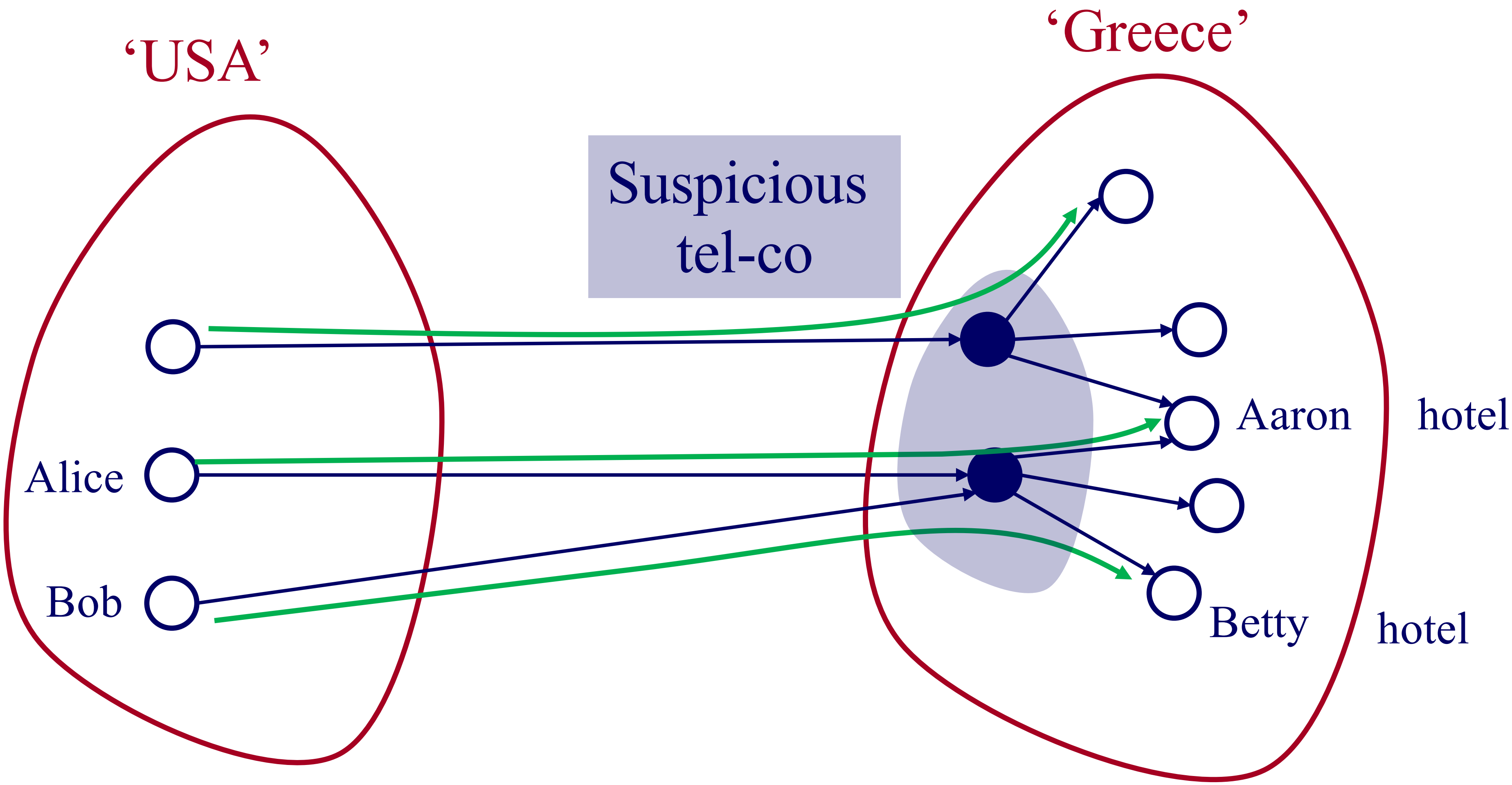
Discovery #1


Group of phone numbers receiving one-second-long calls
in call count -- log10(x+1)
weighted in degree -- log10(x+1)
0
1
2
3
4
5
6
7
10^6
10^5
10^4
10^2
10^1
10^0
log10(N)
confirmed fraudsters
suspicious
suspicious
Hotel
confirmed fraudsters
suspicious
confirmed fraudsters

Q: Why?

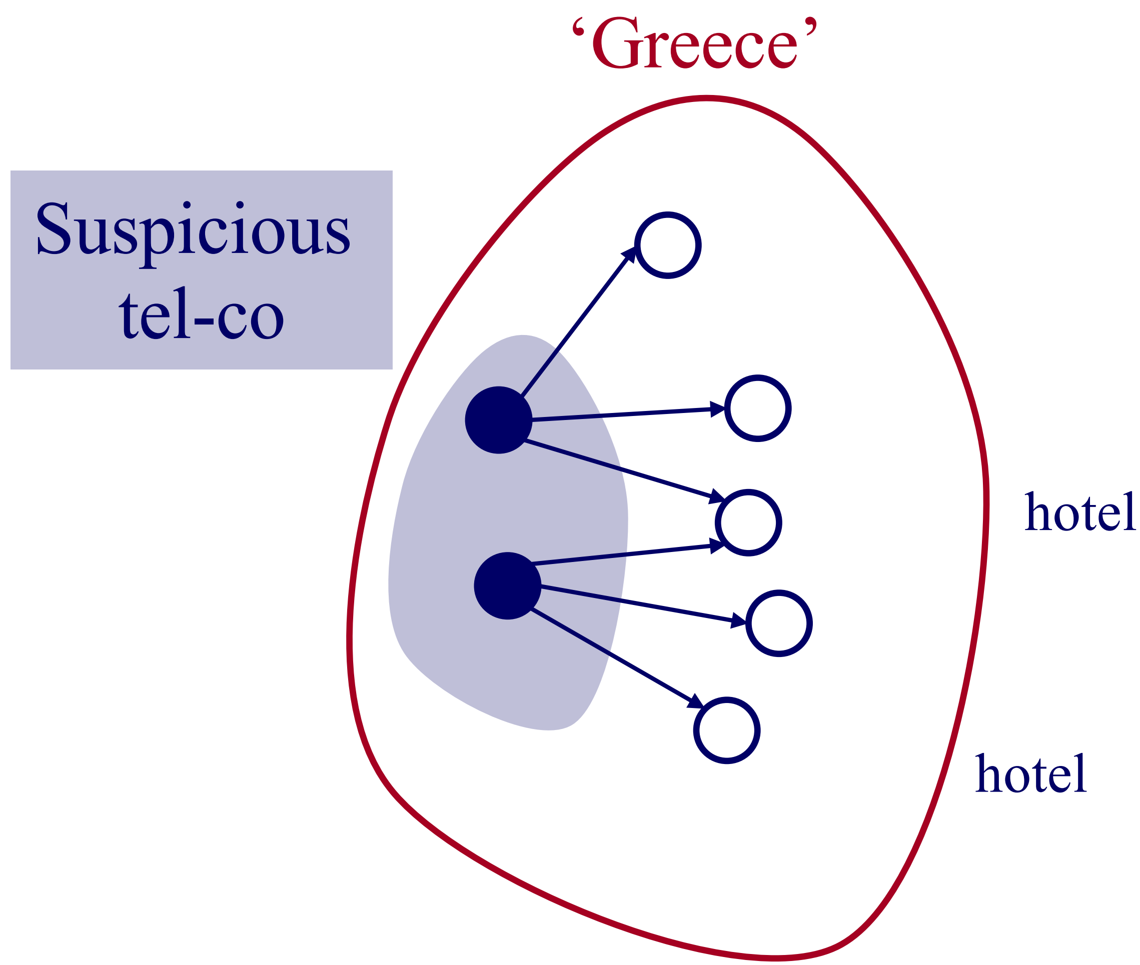
- Q: Why would people call hotel-like numbers, for 1second?

A: ‘international by-pass’



Carnegie Mellon

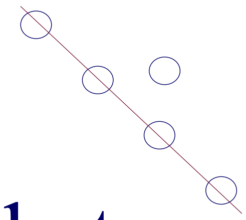
A: ‘international by-pass’



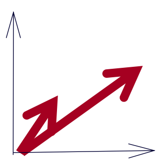
Roadmap

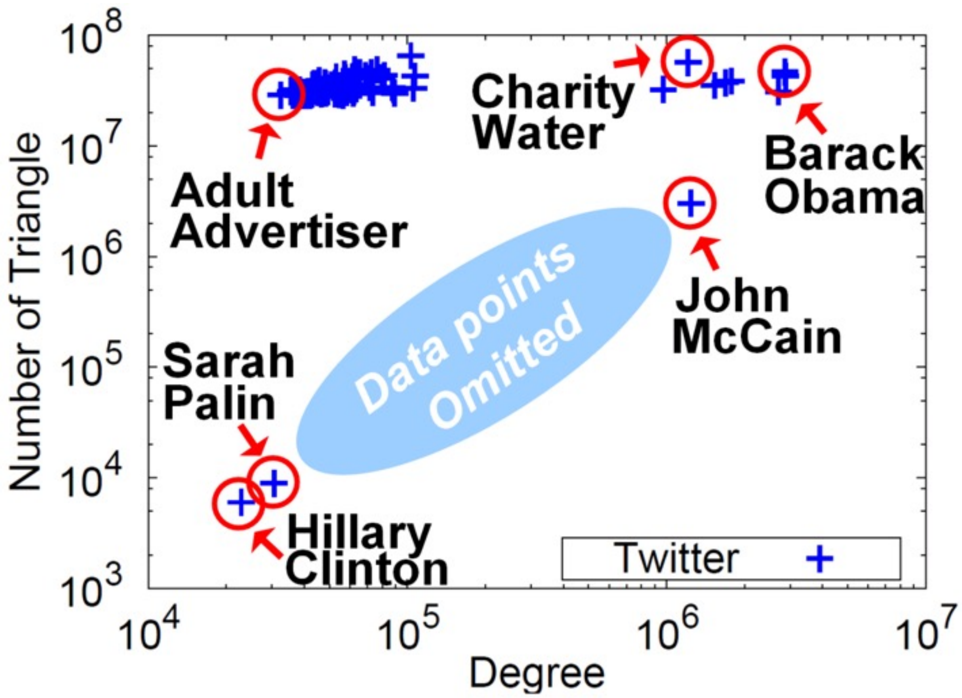
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Visualization
- Conclusions

CONCLUSION#1 – Big data



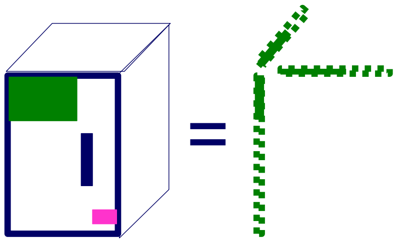
- **Patterns**  **Anomalies**
- **Large** datasets reveal patterns/outliers that are invisible otherwise

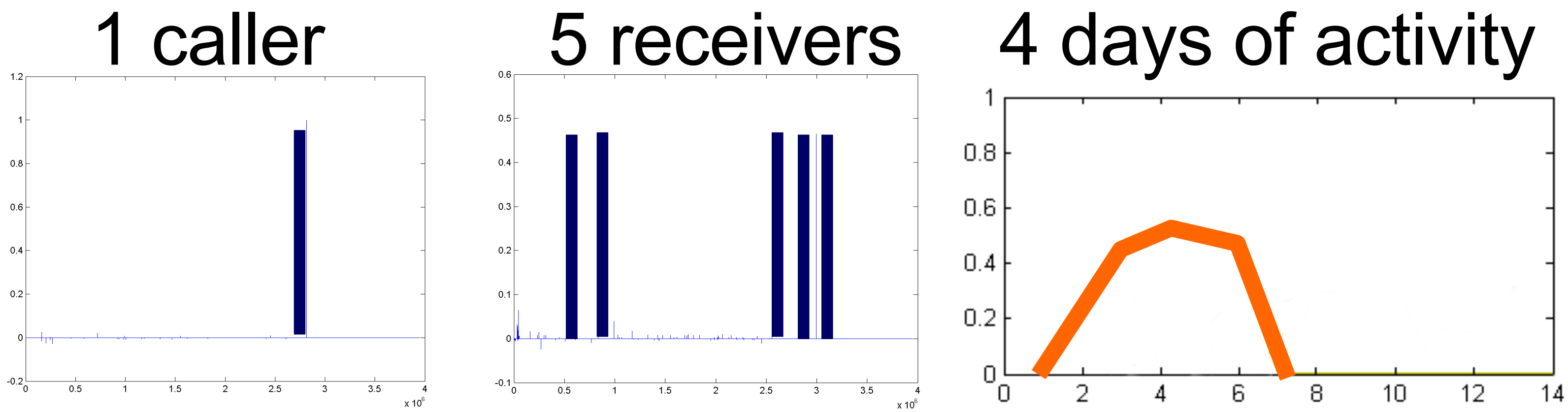




CONCLUSION#2 – tensors

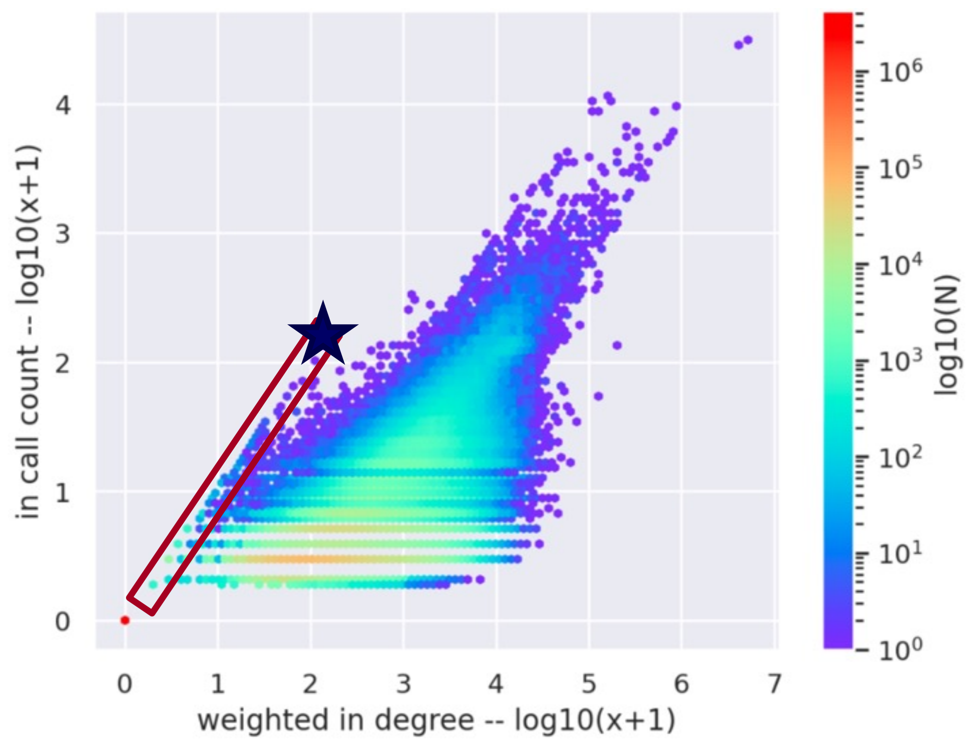
- powerful tool





CONCLUSION#3 - visualization

in-degree

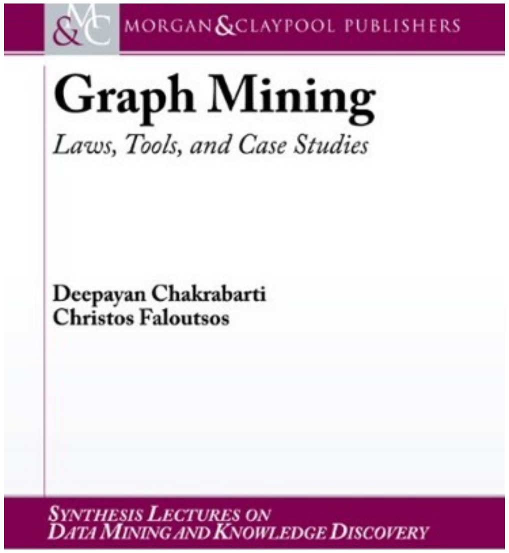


Weighted in-degree (= in-seconds)

Carnegie Mellon

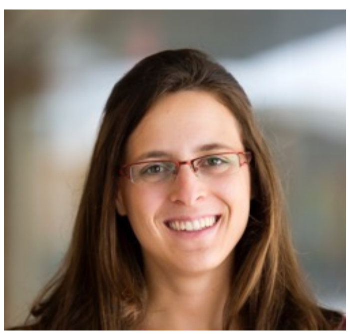
References

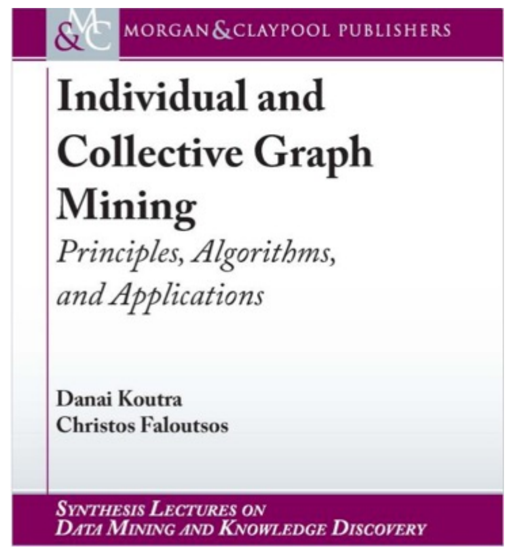
- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
- http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006



References

- Danai Koutra and Christos Faloutsos, *Individual and Collective Graph Mining: Principles, Algorithms, and Applications, Morgan Claypool 2017* (https://doi.org/10.2200/S00796ED1V01Y201708DMK014)



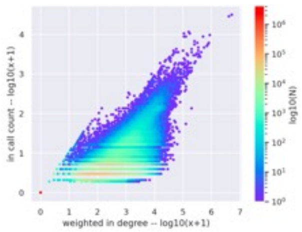


Carnegie Mellon

TgraphSpot: Fast and Effective Anomaly Detection for Time-Evolving Graphs

IEEE BigData, 2022

Mirela Cazzolato[1,2], Saranya Vijayakumar[1], Xinyi Zheng[1], Namyong Park[1], Meng-Chieh Lee[1], Pedro Fidalgo[3,4], Bruno Lages[3], Agma J. M. Traina[2], Christos Faloutsos[1]



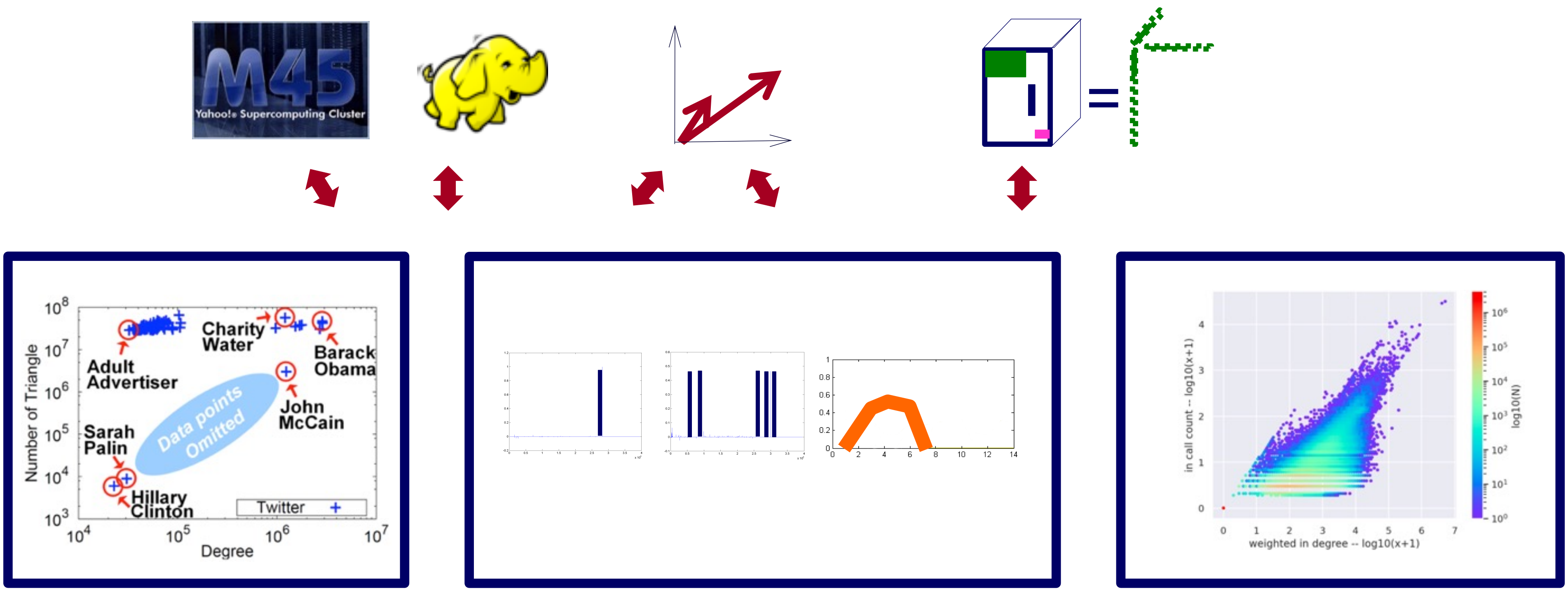
Open source:

`https://github.com/mtcazzolato/tgraph-spot`

Video: https://youtu.be/jI1adN-BQuo?t=1537

TAKE HOME MESSAGE:

Cross-disciplinarity



Carnegie Mellon

Thank you!

Cross-disciplinarity