Carnegie Mellon

Anomaly detection in large graphs

Christos Faloutsos

CMU

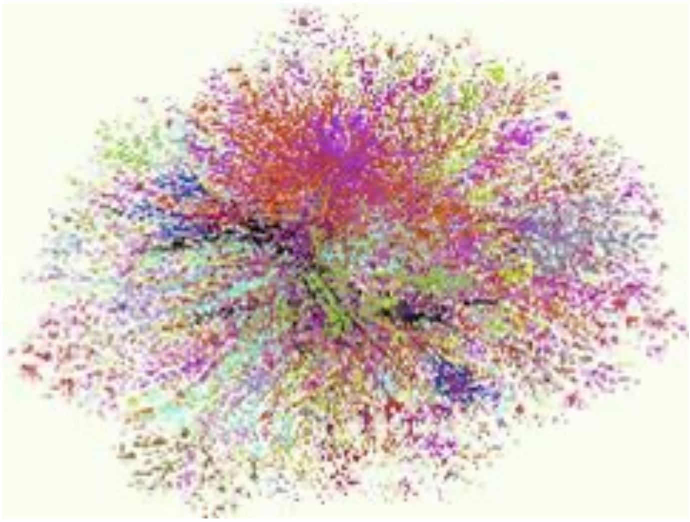
Roadmap

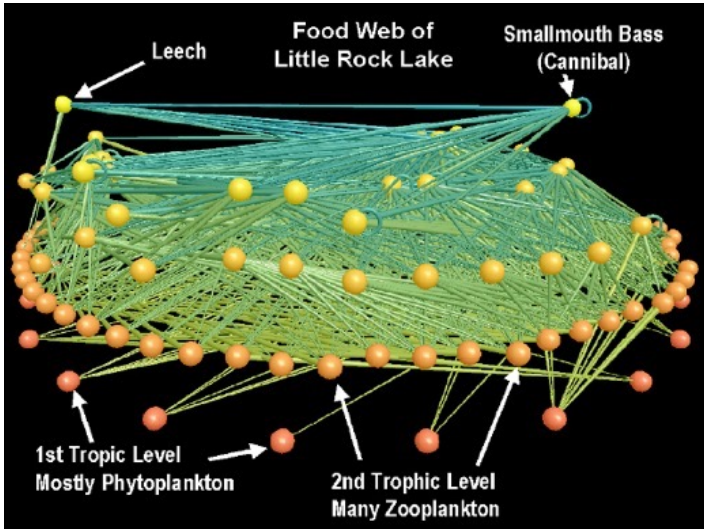
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Conclusions

Graphs - why should we care?

>$10B; ~1B users

Graphs - why should we care?





Internet Map
[lumeta.com]

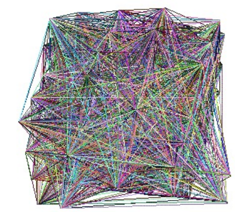
Food Web
[Martinez '91]

Graphs - why should we care?

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

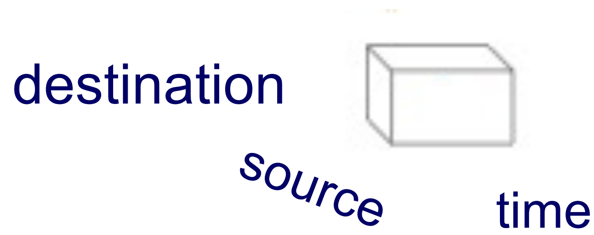
- Many-to-many db relationship -> graph

Motivating problems

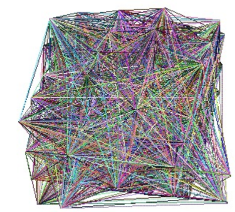


- P1: patterns? Fraud detection?

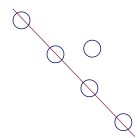
- P2: patterns in time-evolving graphs / tensors



Motivating problems



- P1: patterns? Fraud detection?



Patterns

anomalies

- P2: patterns in time-evolving graphs / tensors

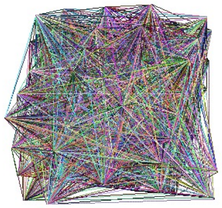
destination

source

time

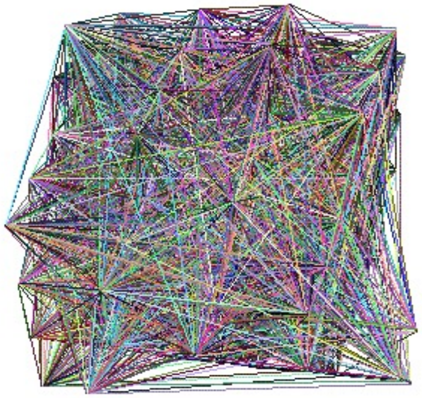
Roadmap

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



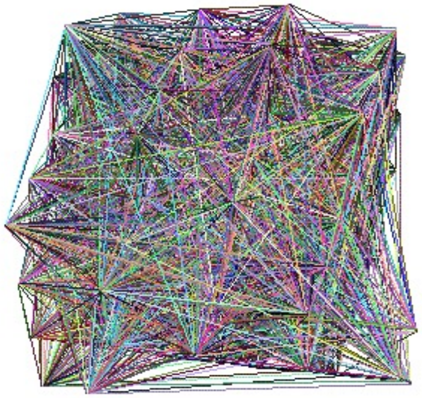
Part 1: Patterns, & fraud detection

Laws and patterns



- Q1: Are real graphs random?

Laws and patterns



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

Solution# S.1

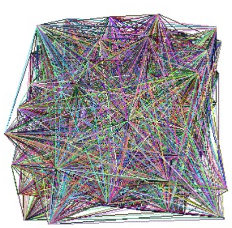
- Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

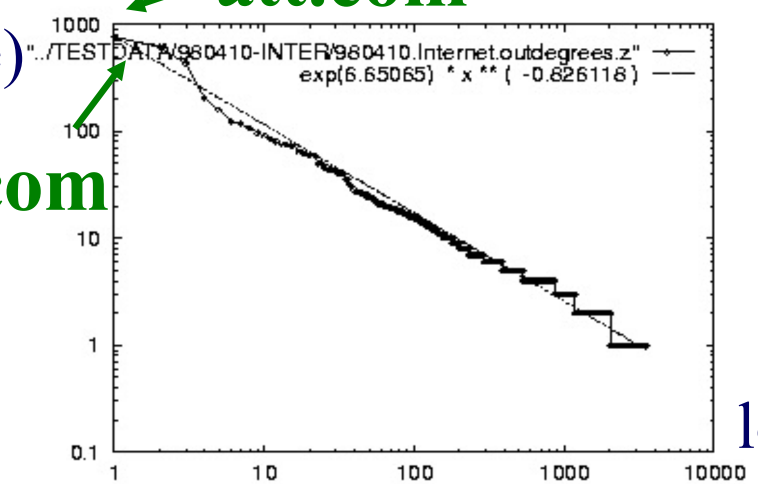
internet domains

att.com

log(degree)

ibm.com



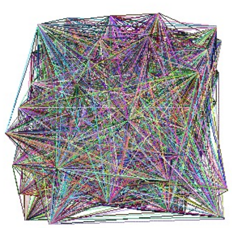


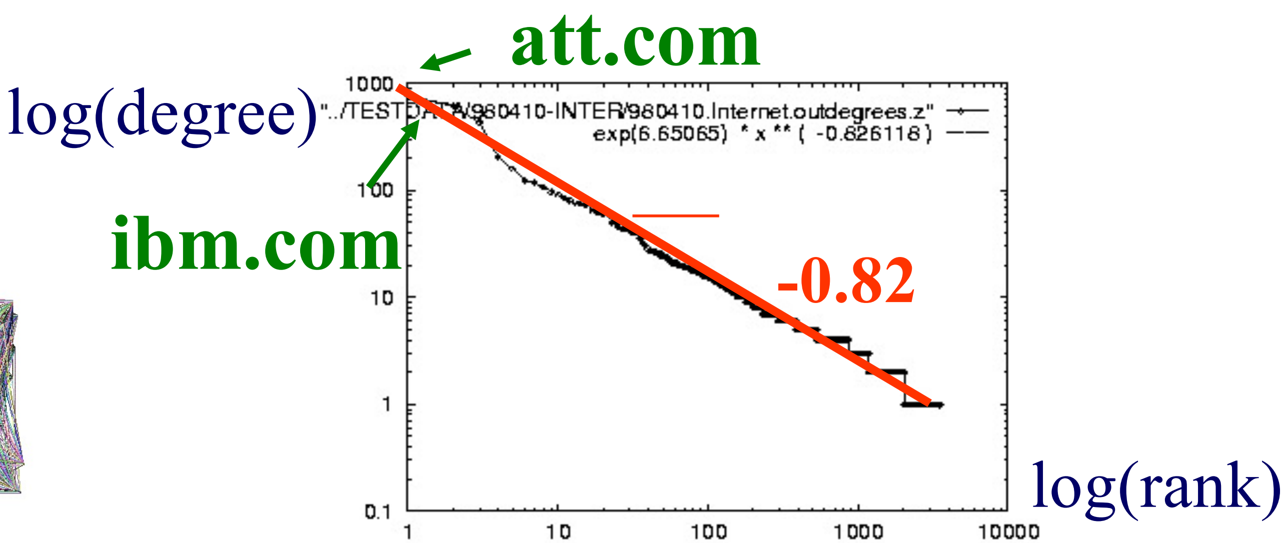
log(rank)

Solution# S.1

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

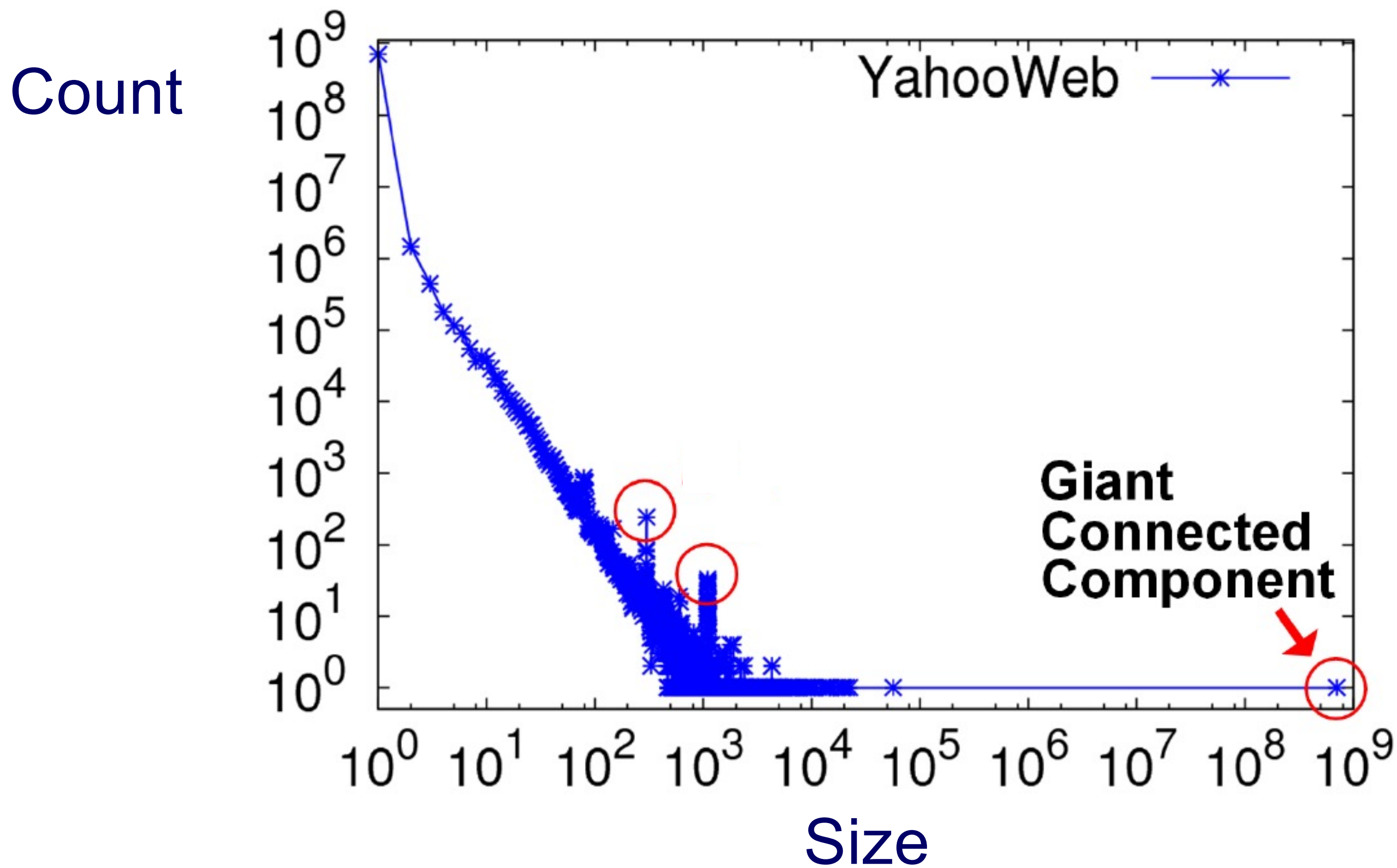
internet domains





S2: connected component sizes

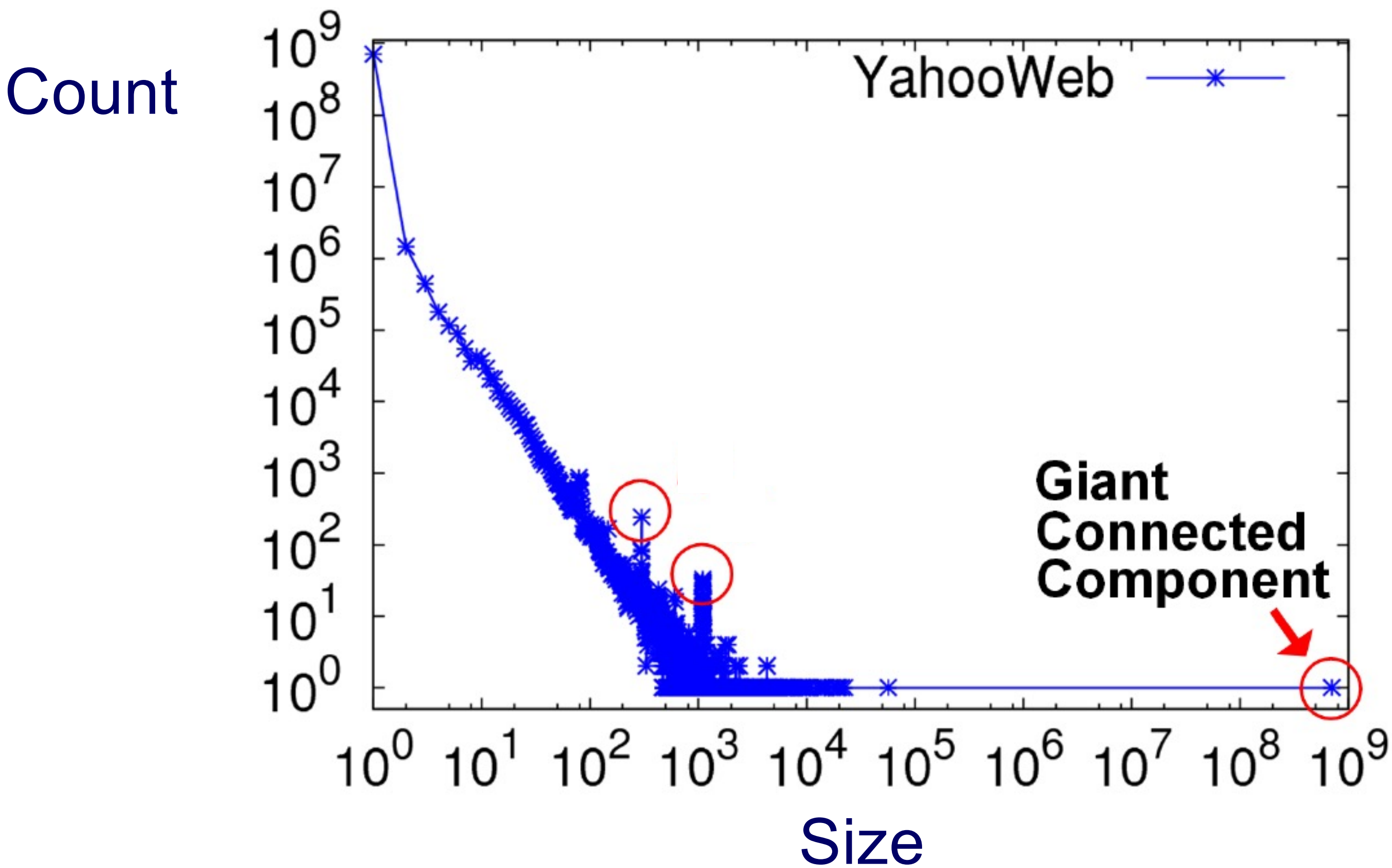
- Connected Components – 4 observations:



1.4B nodes
6B edges

S2: connected component sizes

- Connected Components



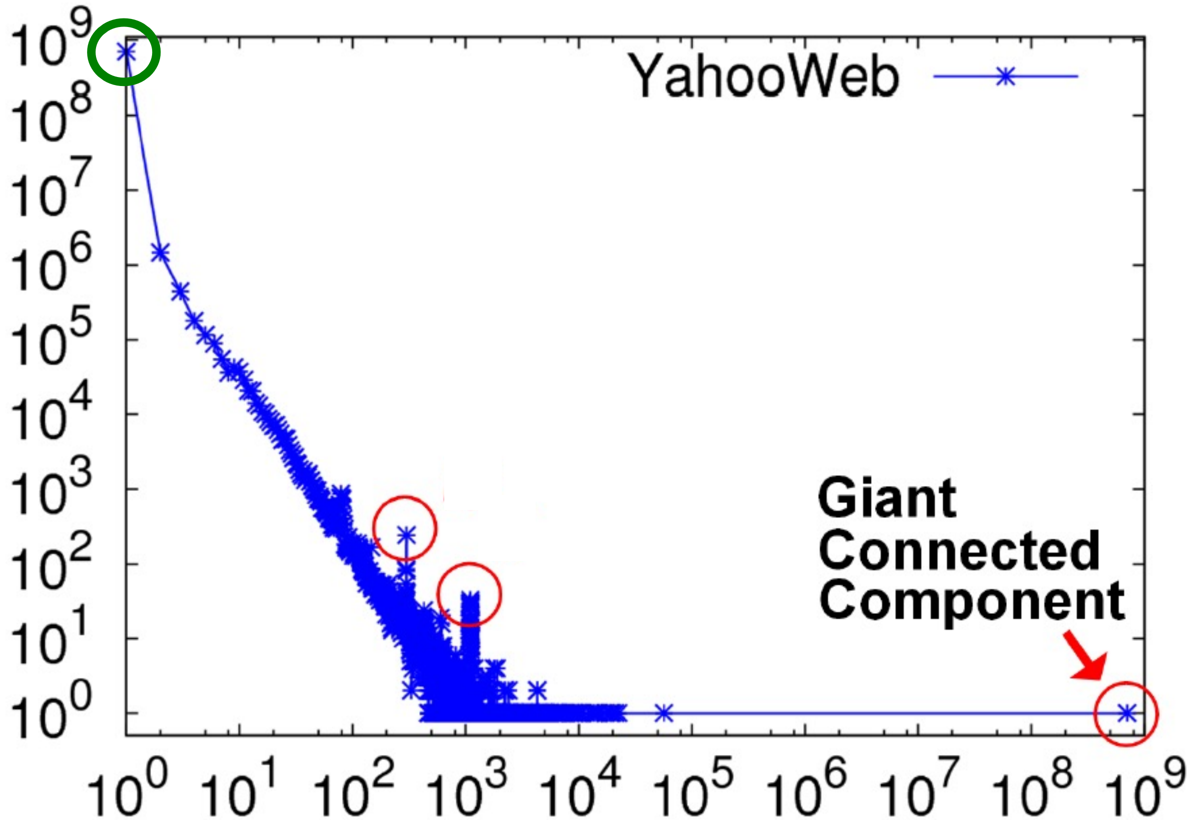
1) 10K x larger than next

S2: connected component sizes

- Connected Components

Count

2) ~0.7B singleton nodes



Size

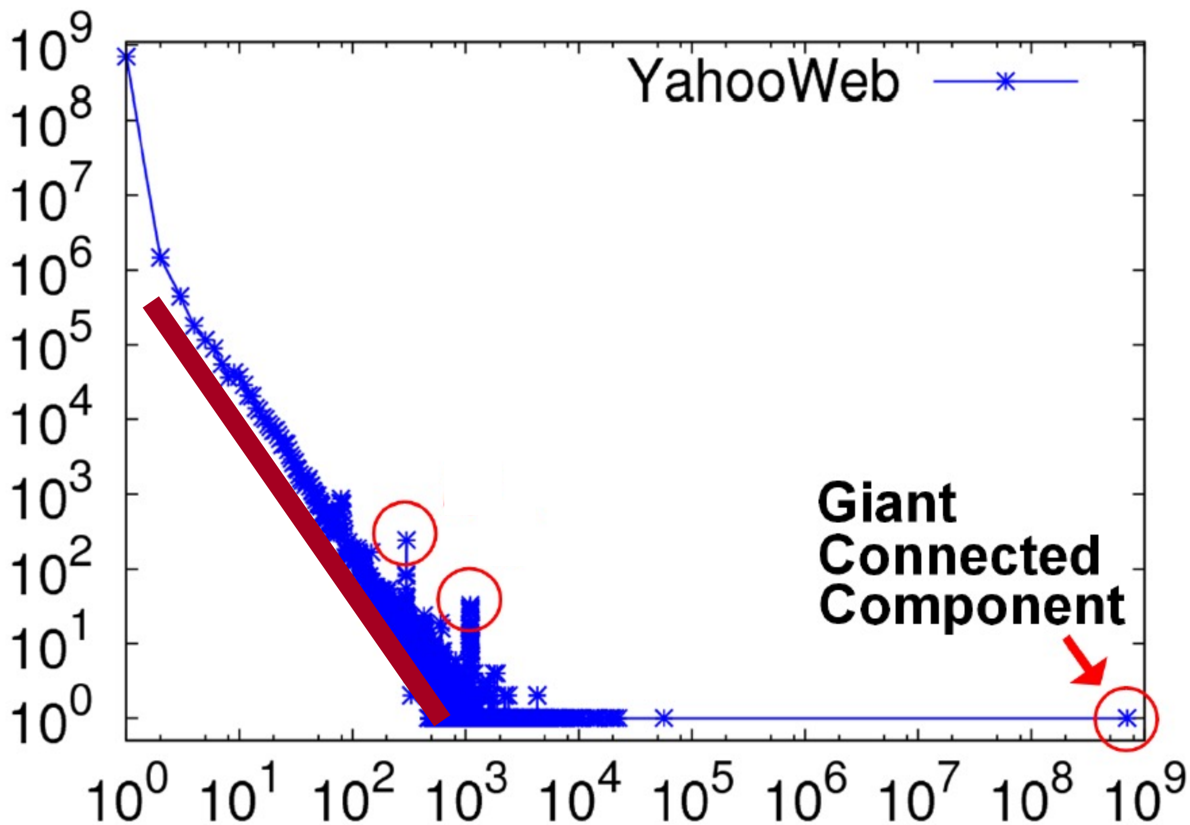
Carnegie Mellon

S2: connected component sizes

- Connected Components

Count

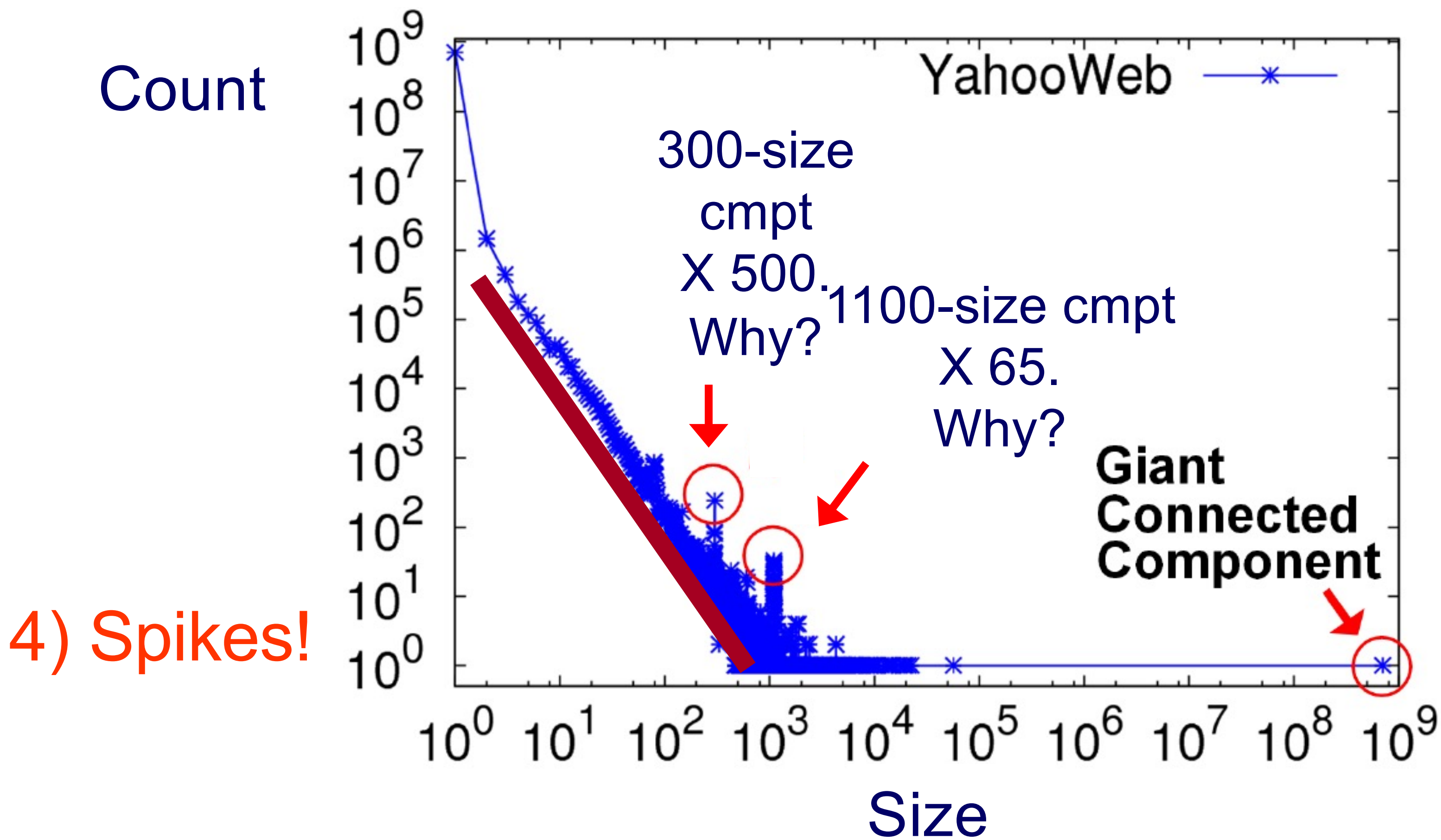
3) SLOPE!



Size

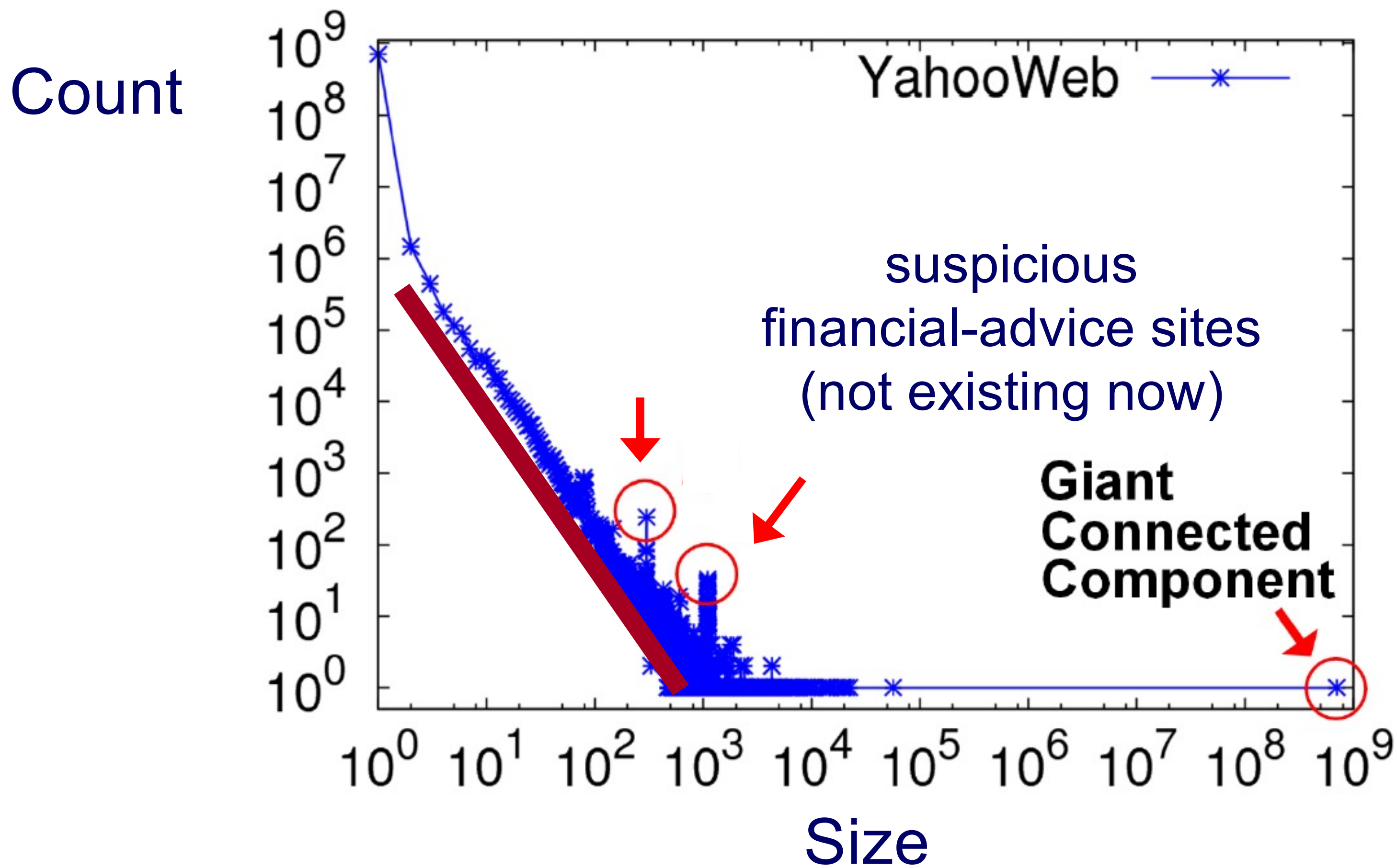
S2: connected component sizes

- Connected Components



S2: connected component sizes

- Connected Components



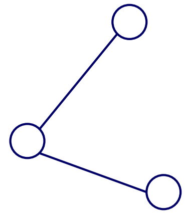
Carnegie Mellon

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

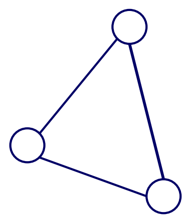
Carnegie Mellon

Solution# S.3: Triangle 'Laws'

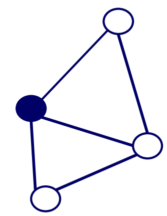


- Real social networks have a lot of triangles

Solution# S.3: Triangle 'Laws'

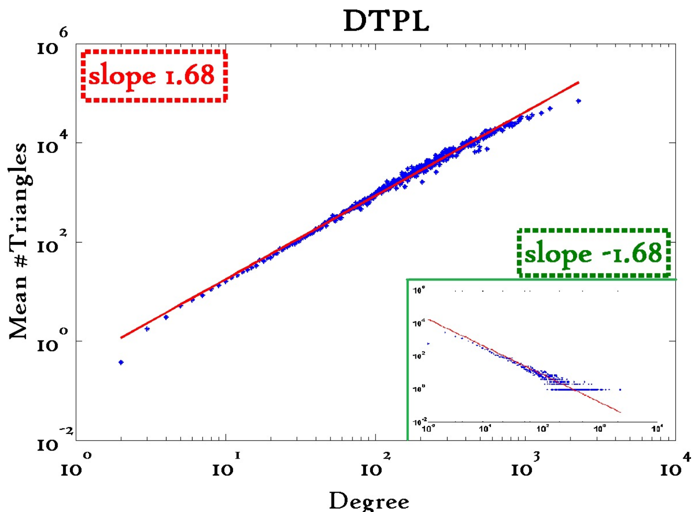


- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?

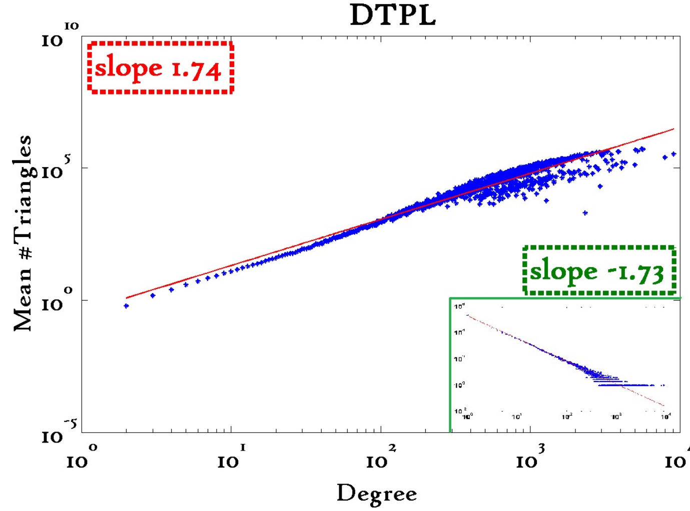


Triangle Law: #S.3
[Tsourakakis ICDM 2008]

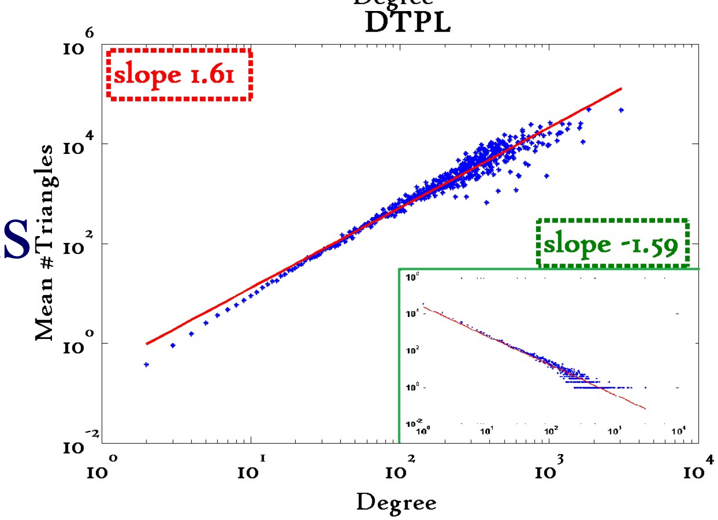
Reuters

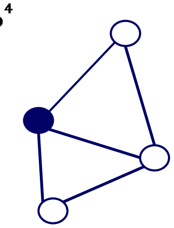


SN



Epinions



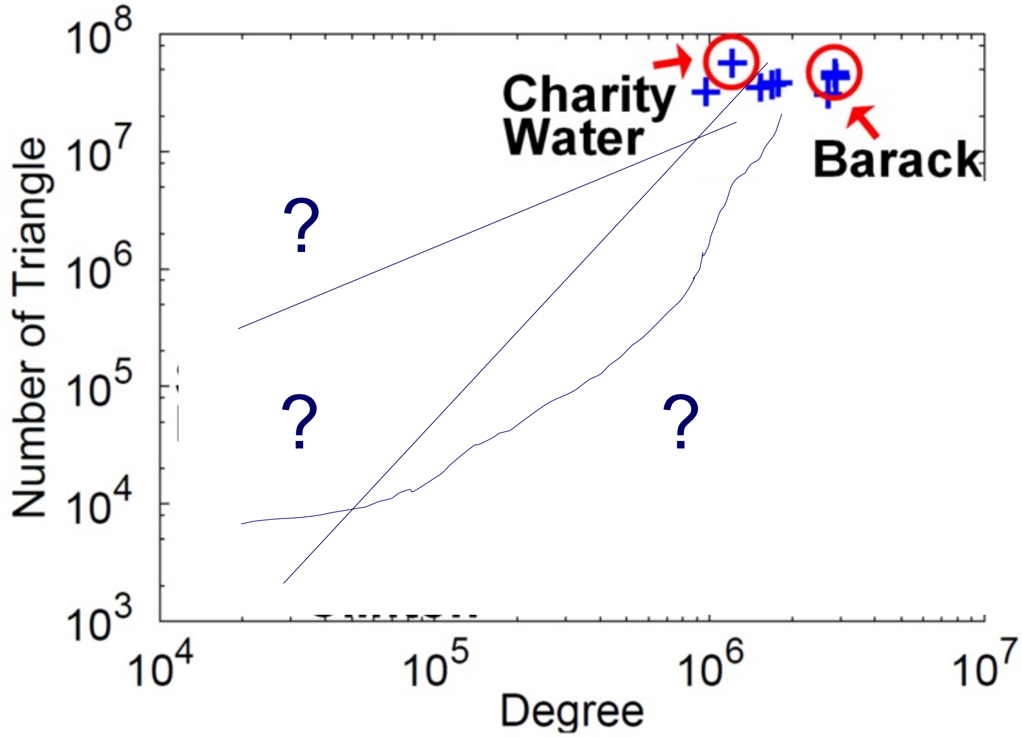


X-axis: degree

Y-axis: mean # triangles

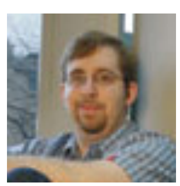
n friends -> $\sim n^{1.6}$ triangles

Triangle counting for large graphs?



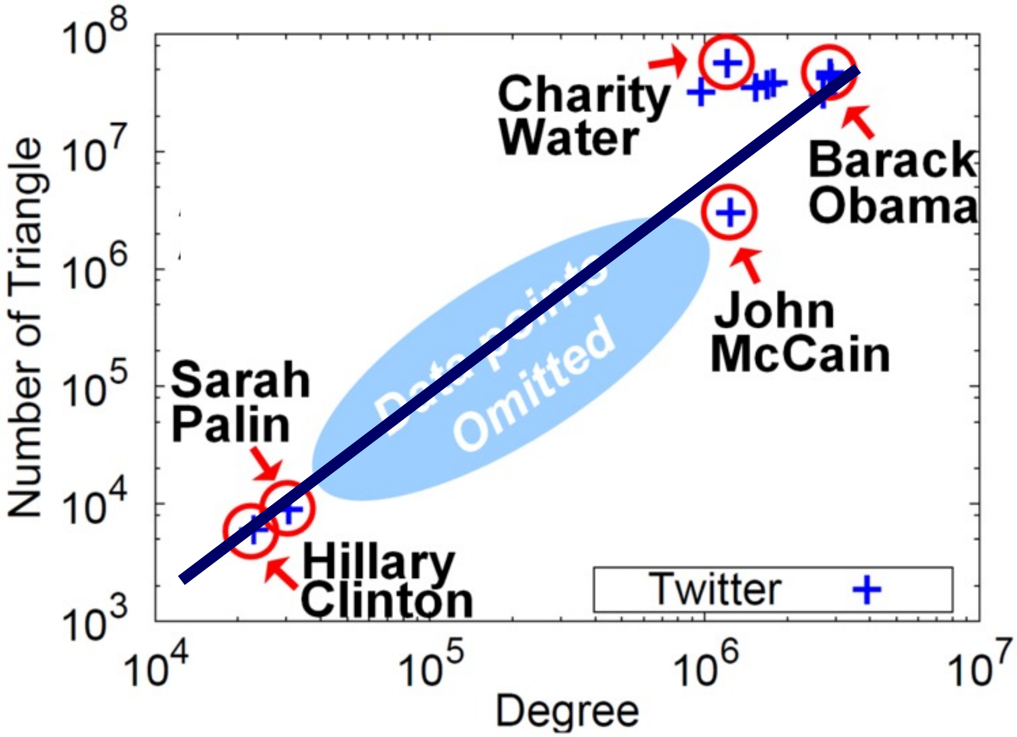
Anomalous nodes in Twitter(~ 3 billion edges)

[U Kang, Brendan Meeder, +, PAKDD’11]



Carnegie Mellon

Triangle counting for large graphs?

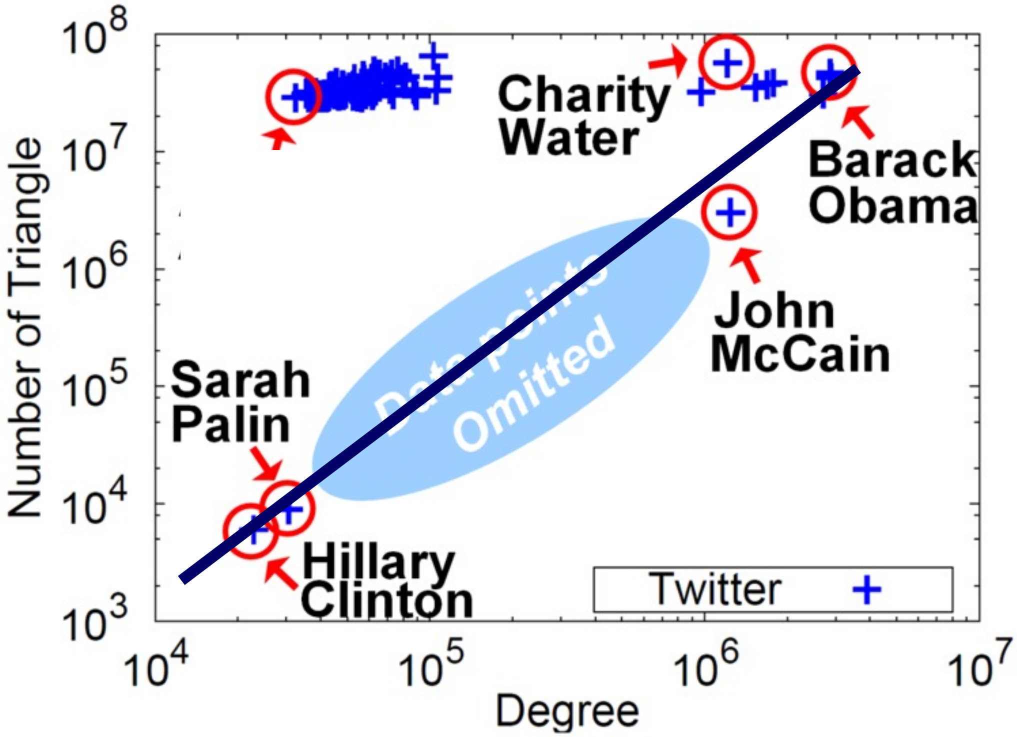


Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

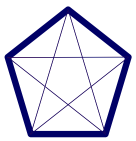
Triangle counting for large graphs?

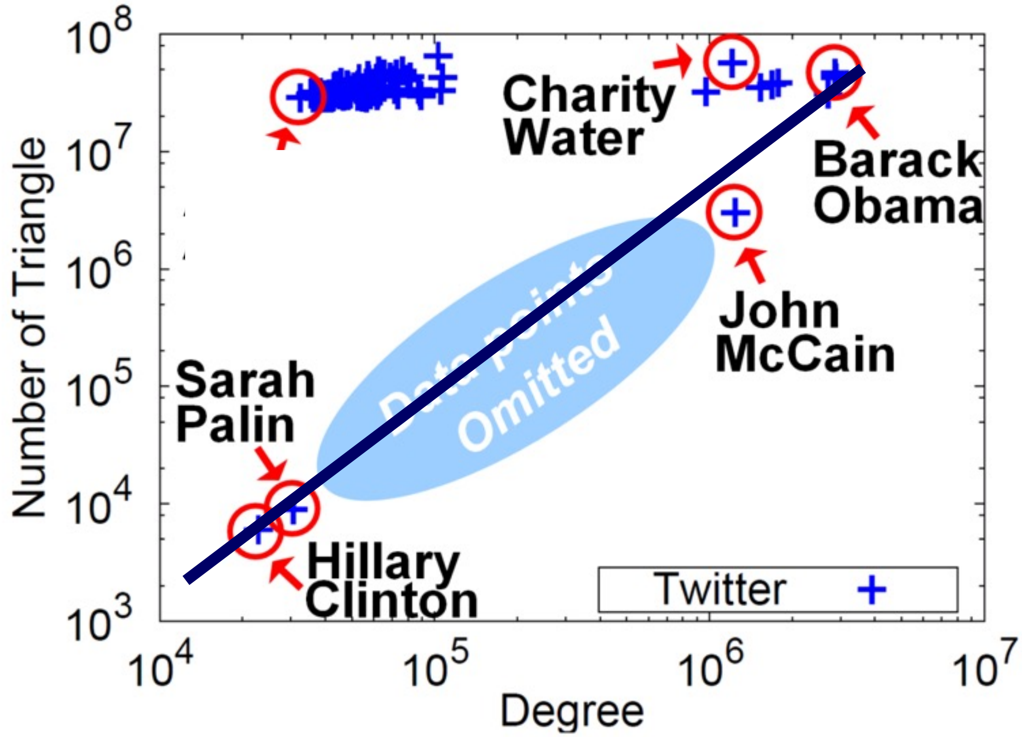


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?

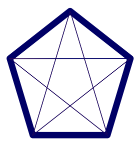


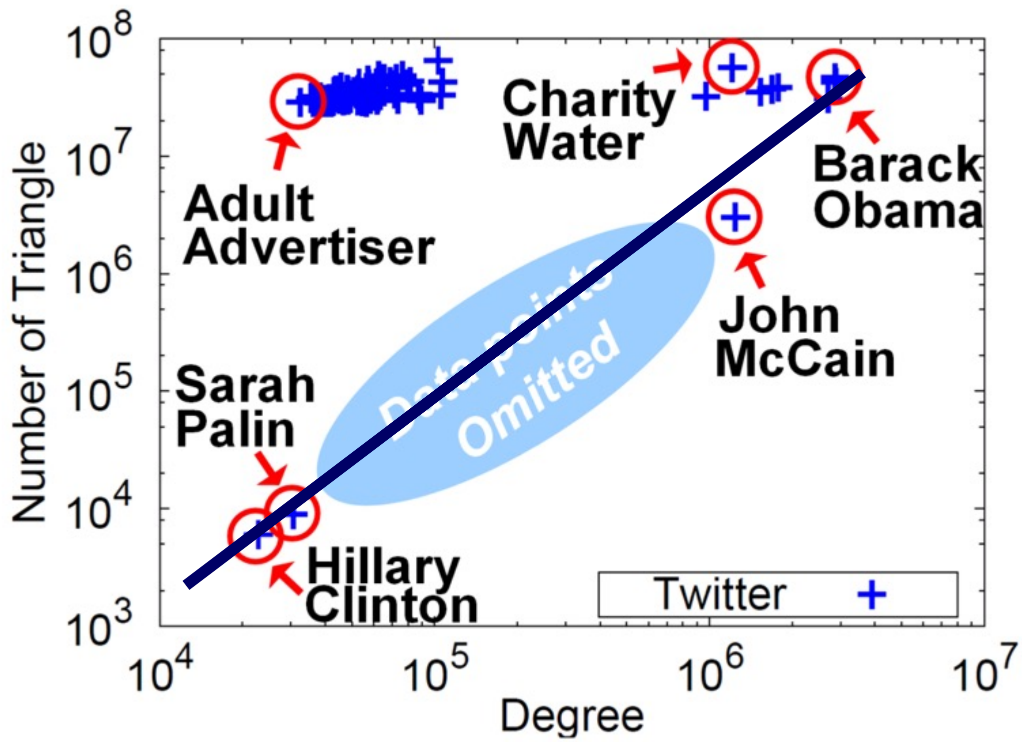


Anomalous nodes in Twitter(~ 3 billion edges)

[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

MORE Graph Patterns

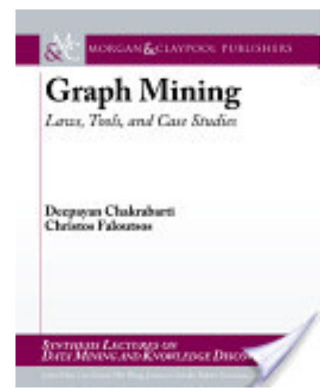
	Unweighted	Weighted
Static	**L01.** Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04] **L02.** Triangle Power Law (TPL) [Tsourakakis `08] **L03.** Eigenvalue Power Law (EPL) [Siganos et al. `03] **L04.** Community structure [Flake et al. `02, Girvan and Newman `02]	**L10.** Snapshot Power Law (SPL) [McGlohon et al. `08]
Dynamic	**L05.** Densification Power Law (DPL) [Leskovec et al. `05] **L06.** Small and shrinking diameter [Albert and Barabási `99, Leskovec et al. `05] **L07.** Constant size $2^{nd}$ and $3^{rd}$ connected components [McGlohon et al. `08] **L08.** Principal Eigenvalue Power Law (λ_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

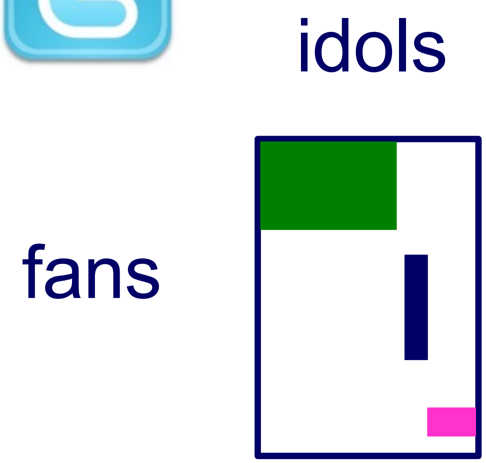
Roadmap

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

Patterns  anomalies

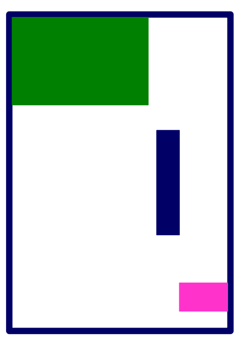
How to find ‘suspicious’ groups?

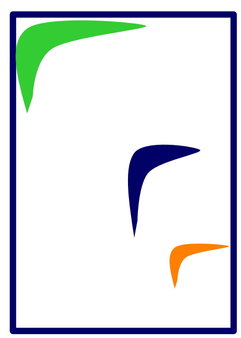
- ‘blocks’ are normal, right?



Except that:

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

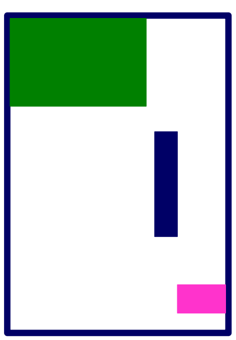


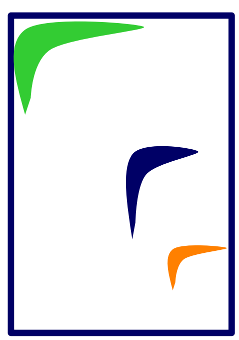


Except that:

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

Q: Can we spot blocks, easily?



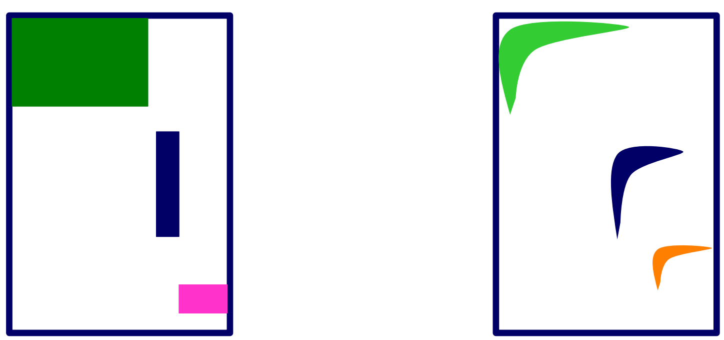


Except that:

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

Q: Can we spot blocks, easily?

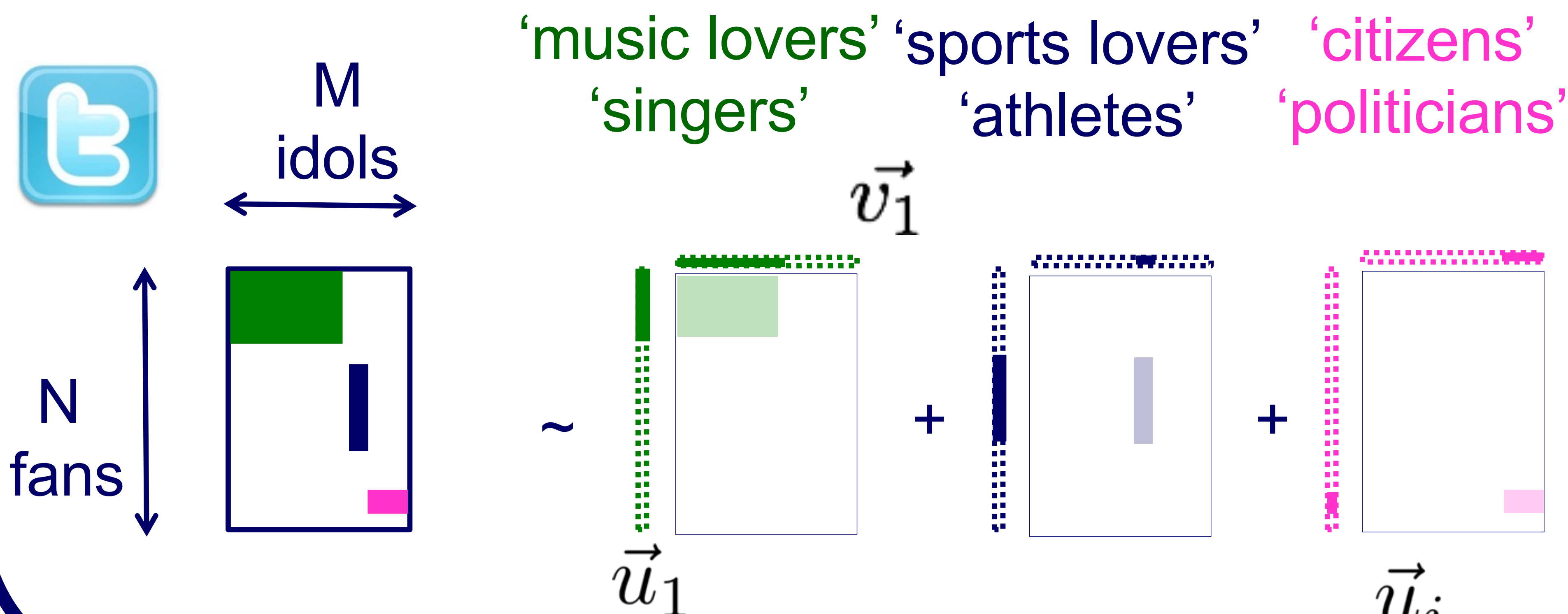
A: Silver bullet: SVD!



DETAILS

Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks

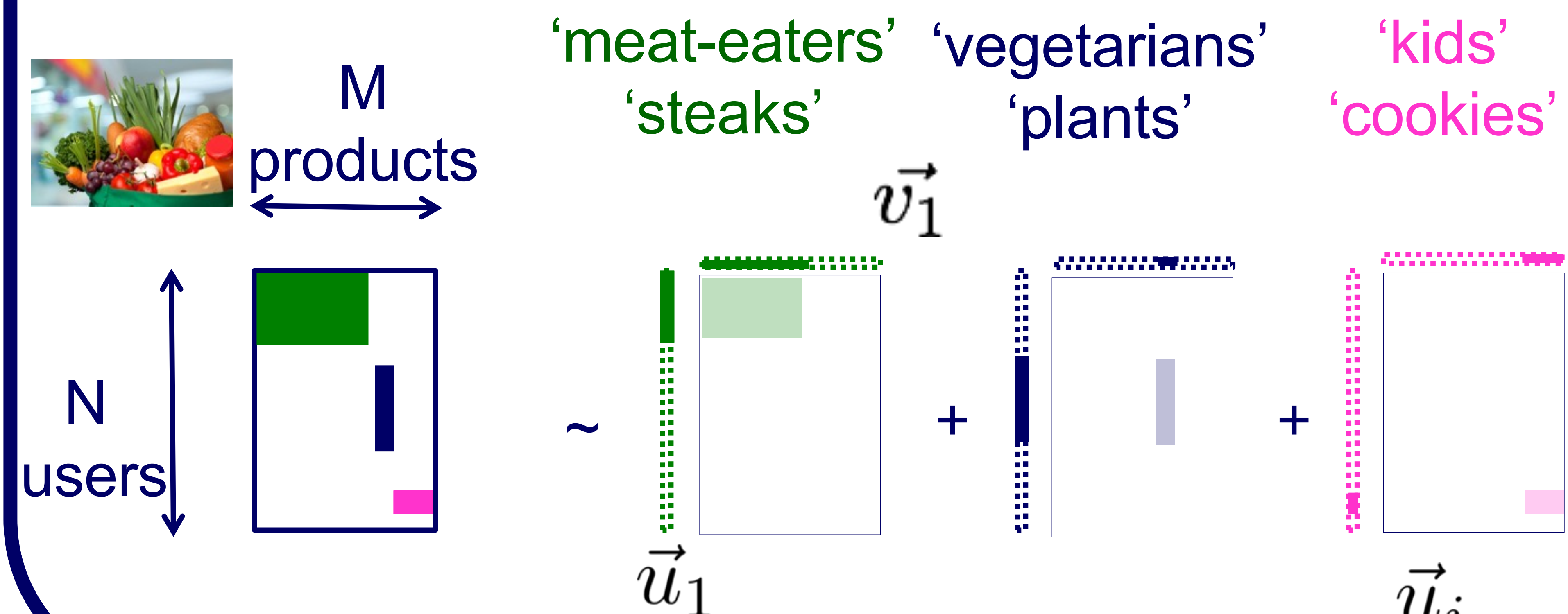


Carnegie Mellon

DETAILS

Crush intro to SVD

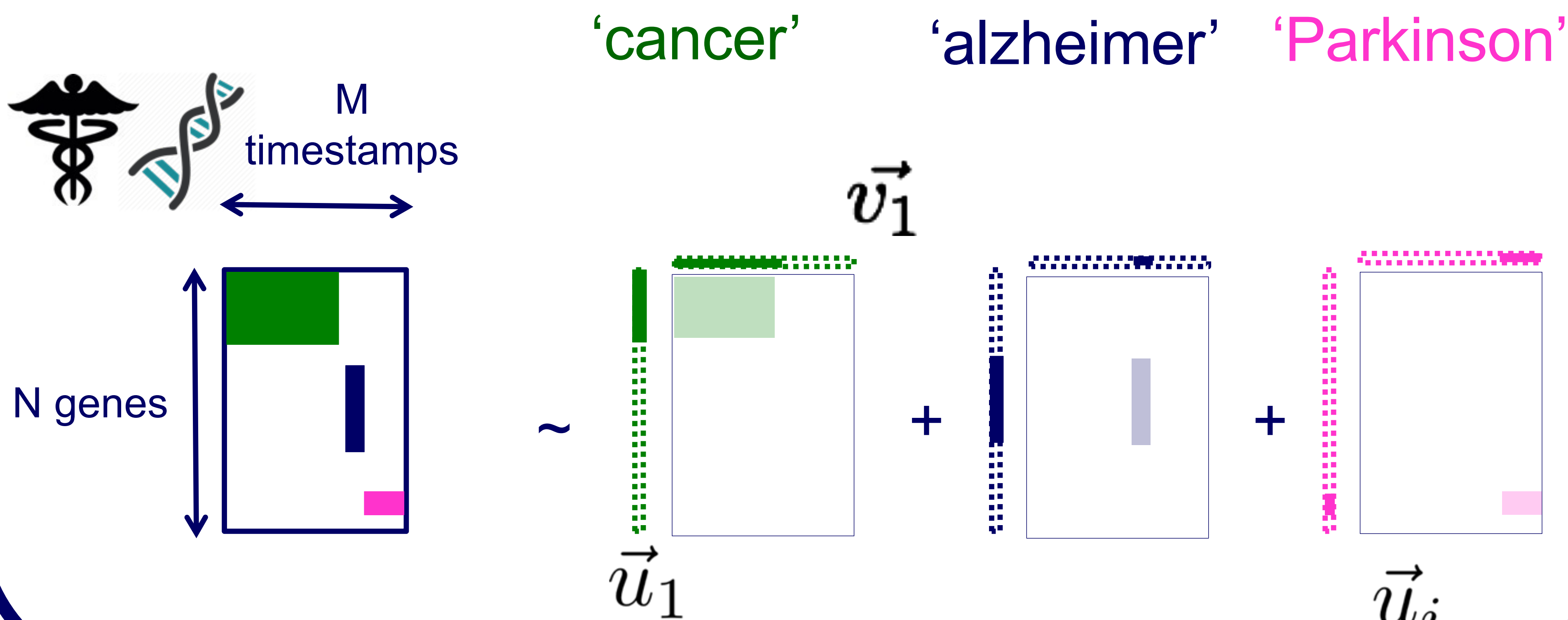
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

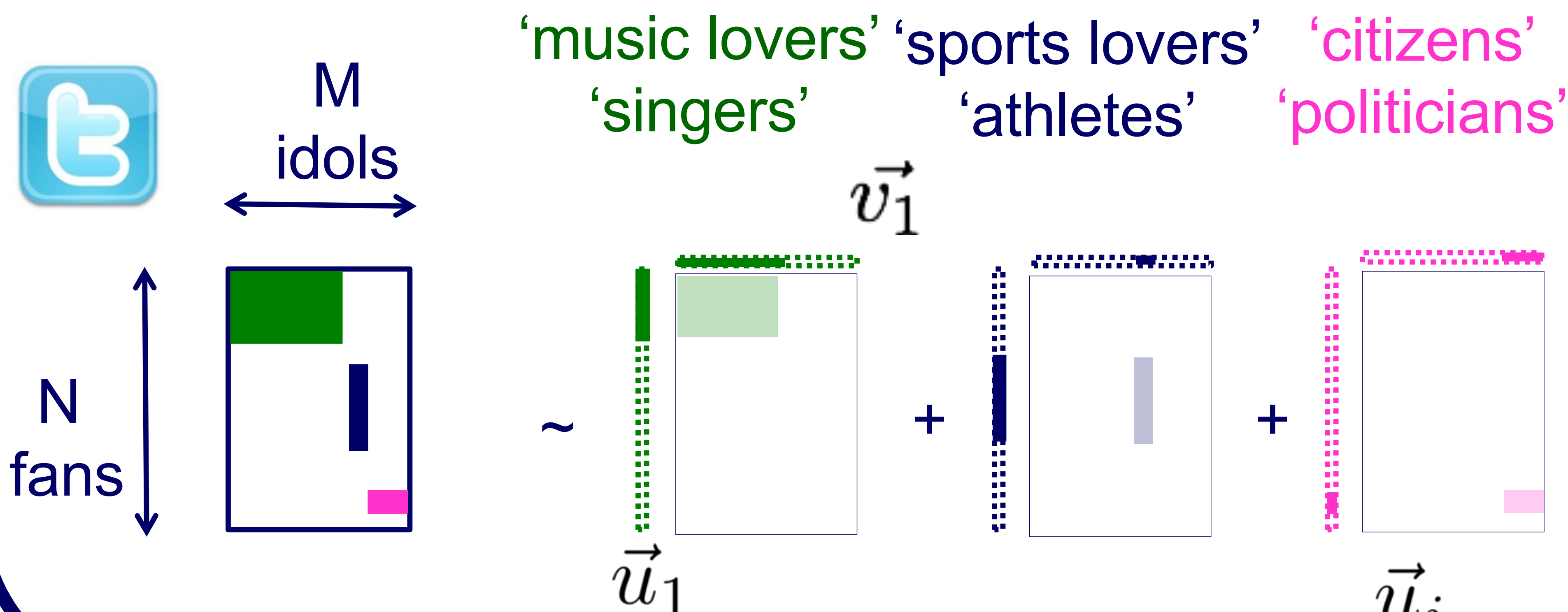
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

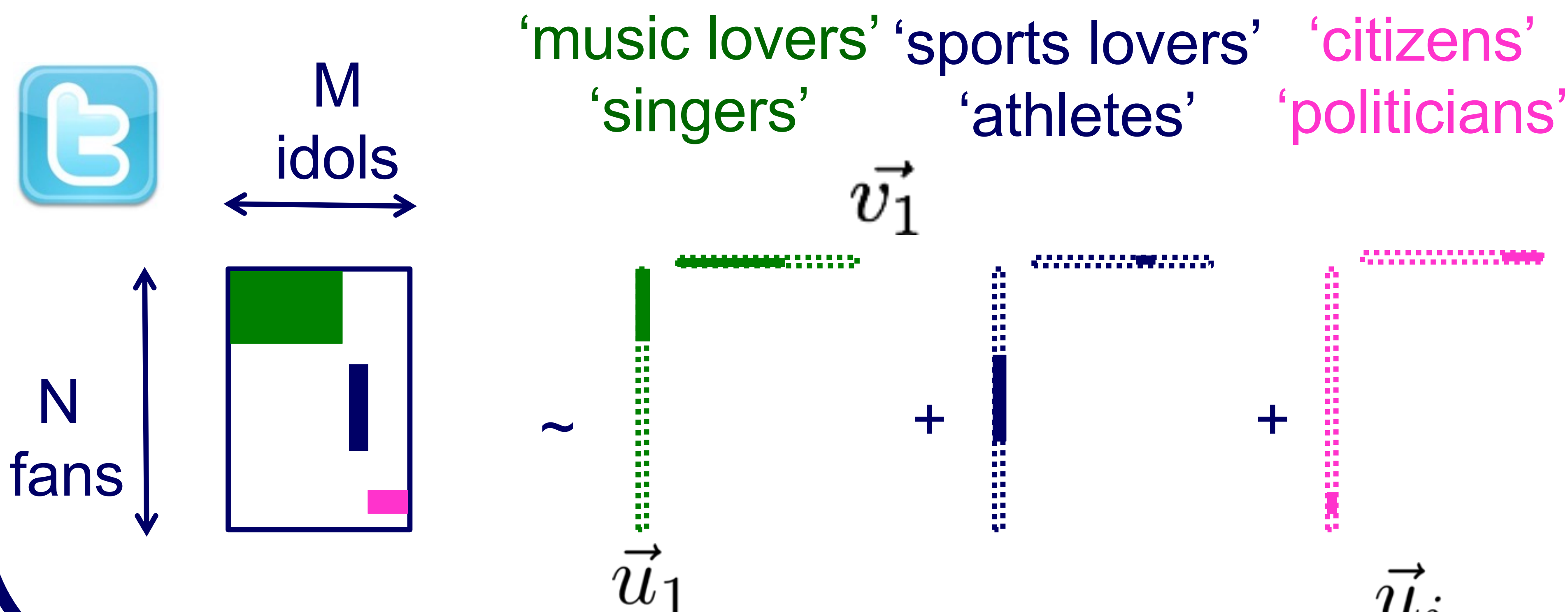
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

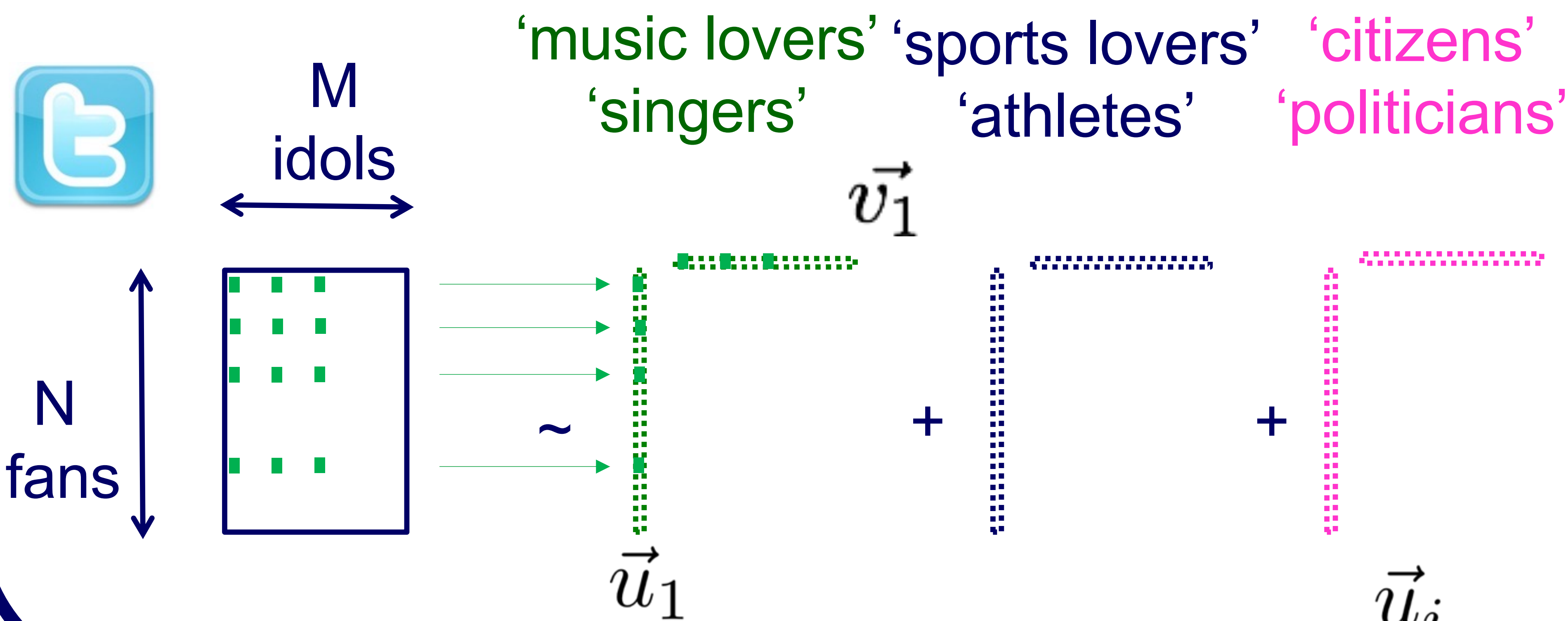
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

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



Carnegie Mellon

Inferring Strange Behavior from Connectivity Pattern in Social Networks

PAKDD'14

Meng Jiang, Peng Cui, Shiqiang Yang (Tsinghua)

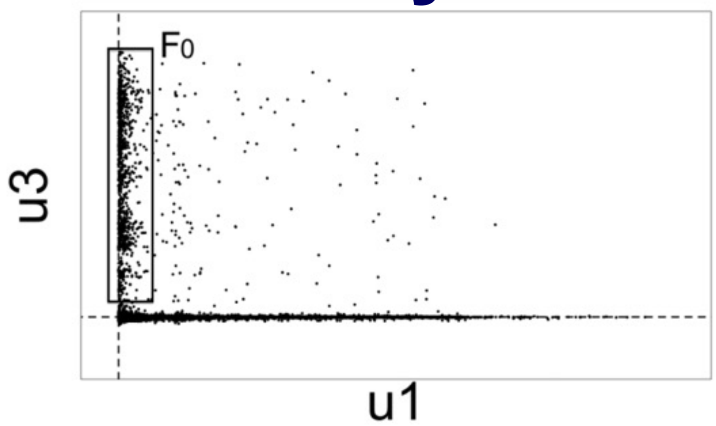
Alex Beutel, Christos Faloutsos (CMU)

Dataset

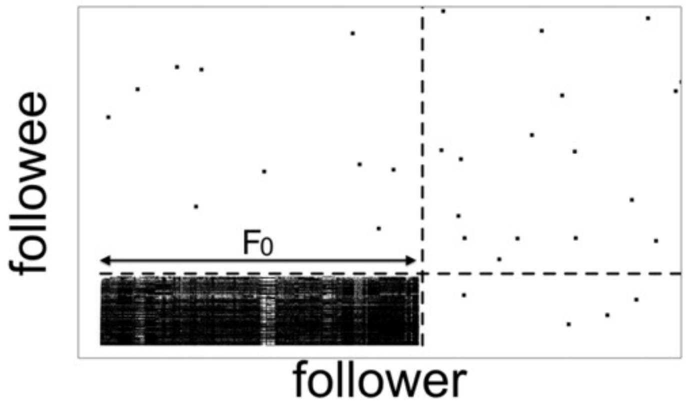
- Tencent Weibo
- 117 million nodes (with profile and UGC data)
- 3.33 billion directed edges

Real Data

"Rays"

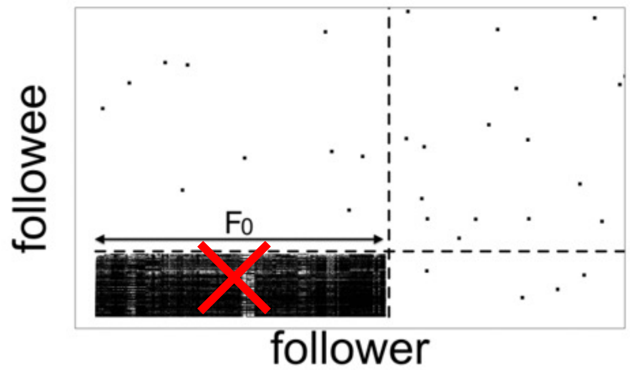


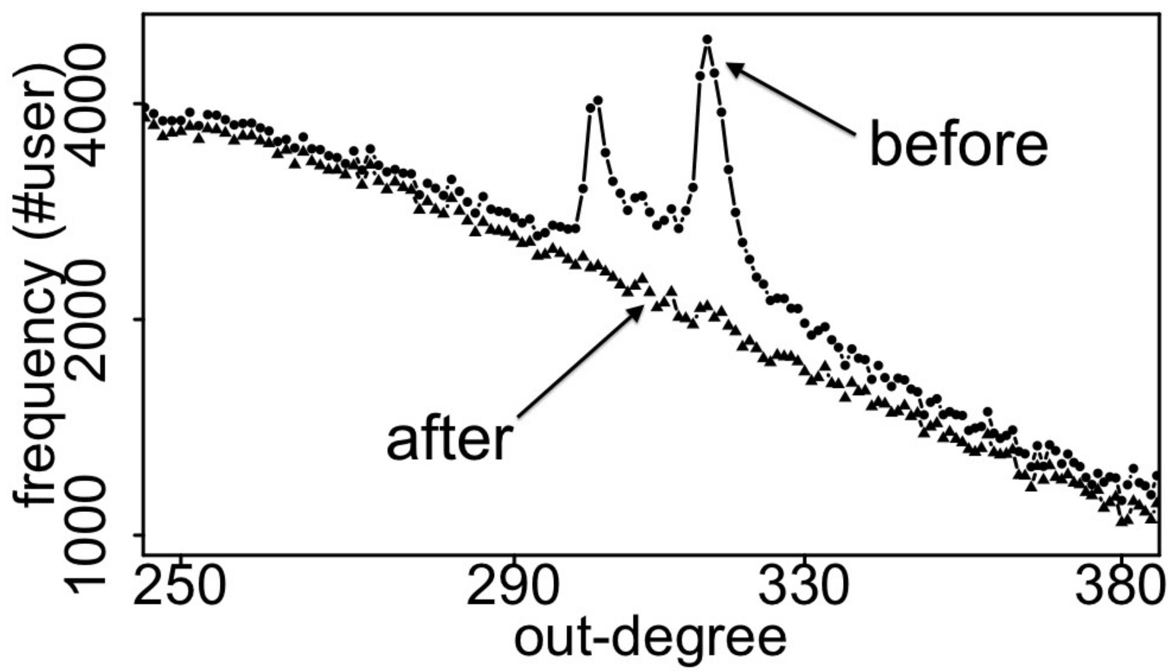
"Block"



Real Data

- Spikes on the out-degree distribution





Roadmap

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

Patterns  anomalies

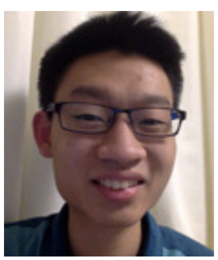
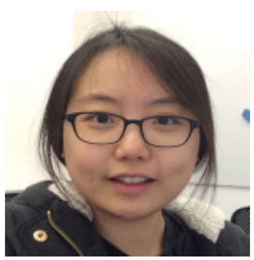
Carnegie Mellon

Knowledge Discovery and Data Mining (KDD) 2016

FRAUDAR: Bounding Graph Fraud in the Face of Camouflage

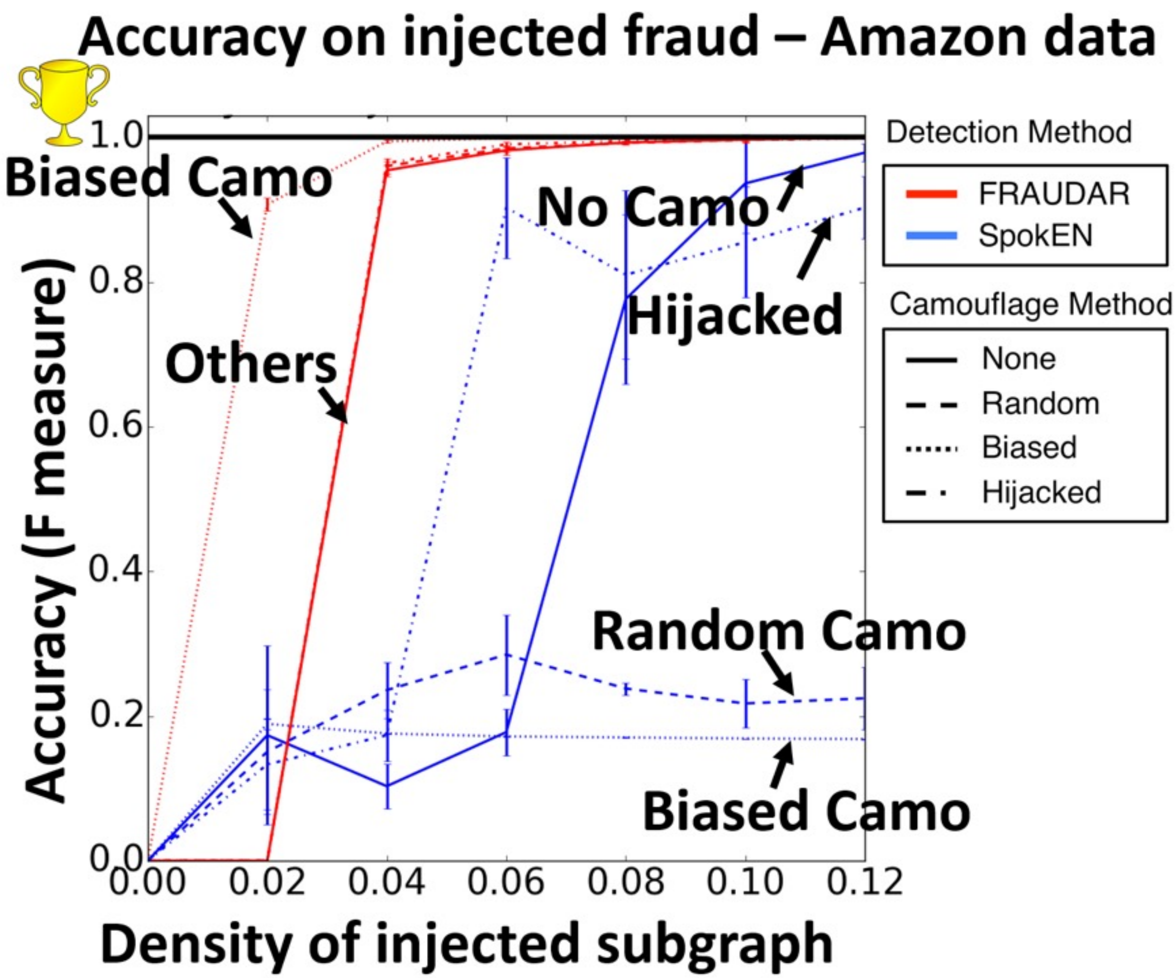
Bryan Hooi, Hyun Ah Song, Alex Beutel,
Neil Shah, Kijung Shin, Christos Faloutsos

Carnegie Mellon University

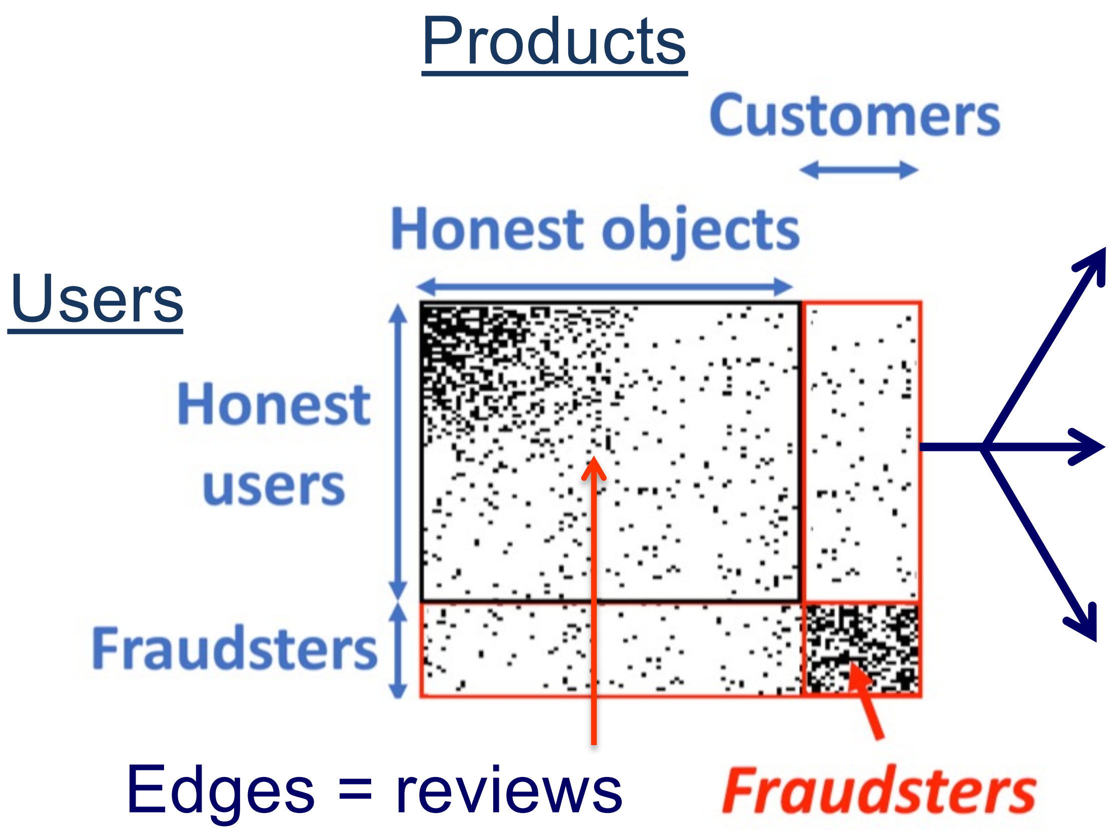
Carnegie Mellon

Experiments: Amazon data



Detecting Review Spam

Many existing methods detect dense subgraphs.



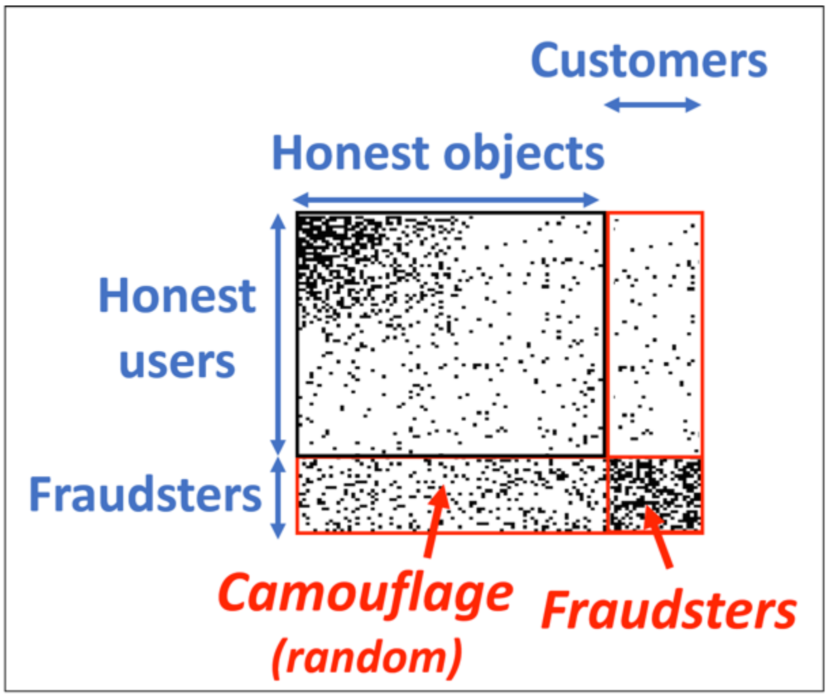
Spectral methods

Clustering

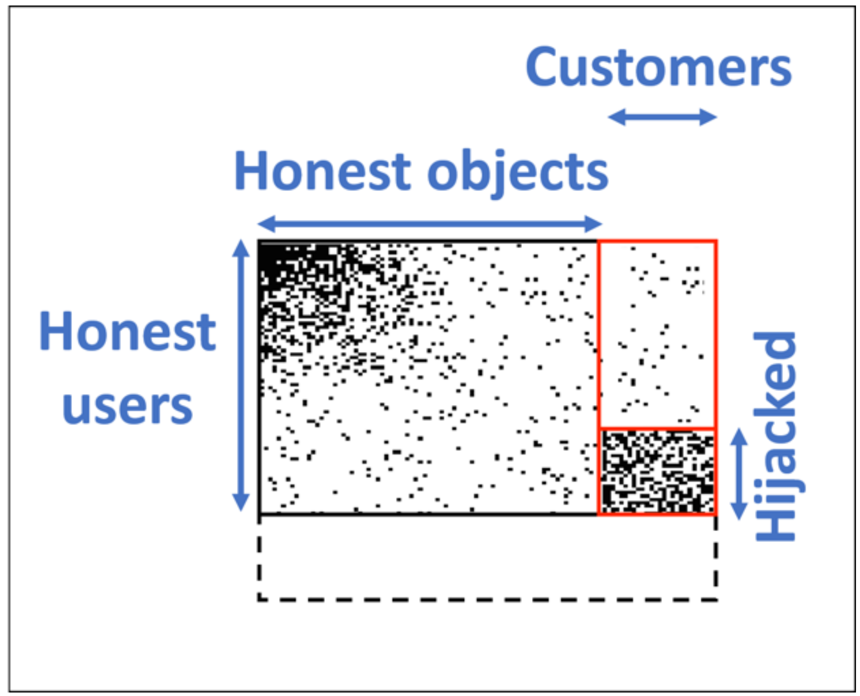
Belief propagation

Evading Detection

Attackers use ***camouflage*** to evade detection.



Random camouflage

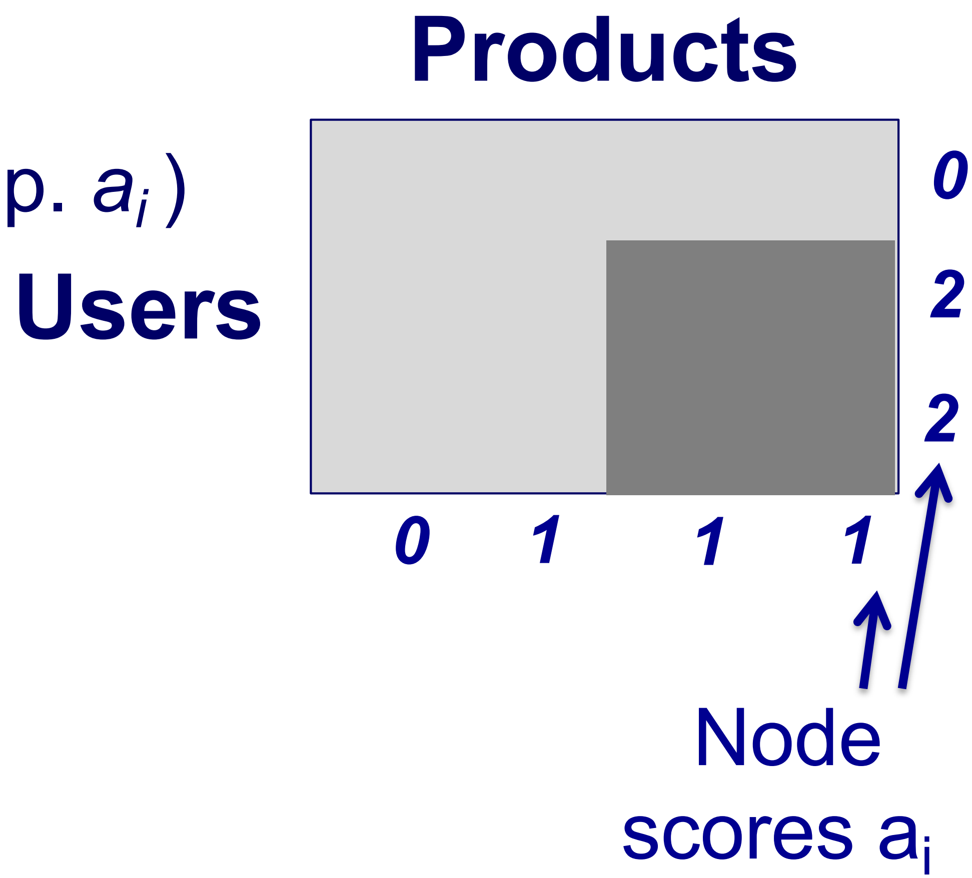


Hijacked user accounts

Problem Definition

Given:

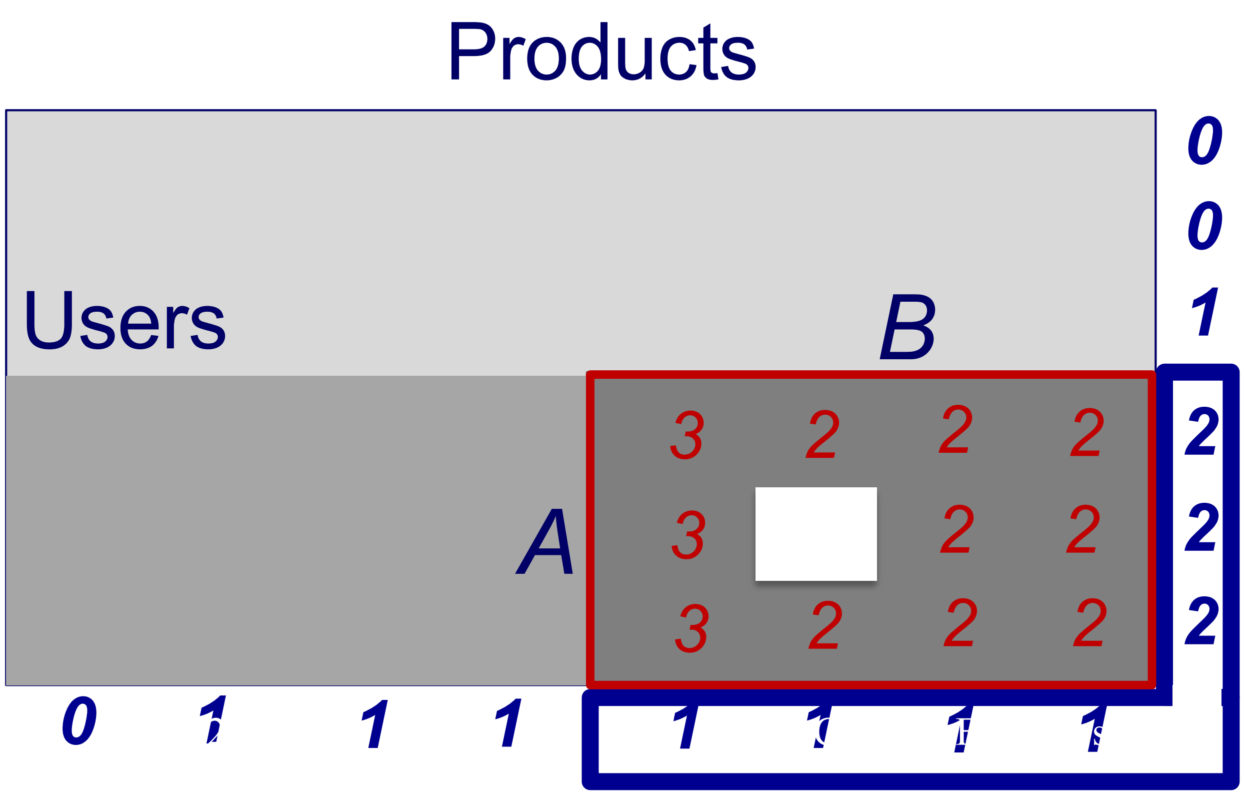
- Bipartite graph between users and products
- (optional: prior node susp. a_i)



Dfn: Average suspiciousness

`Average suspiciousness' g(A,B) =

$$\frac{\text{(sum of node susp.)} + \text{(sum of edge susp.)}}{|A| + |B|}$$



Node scores a_i

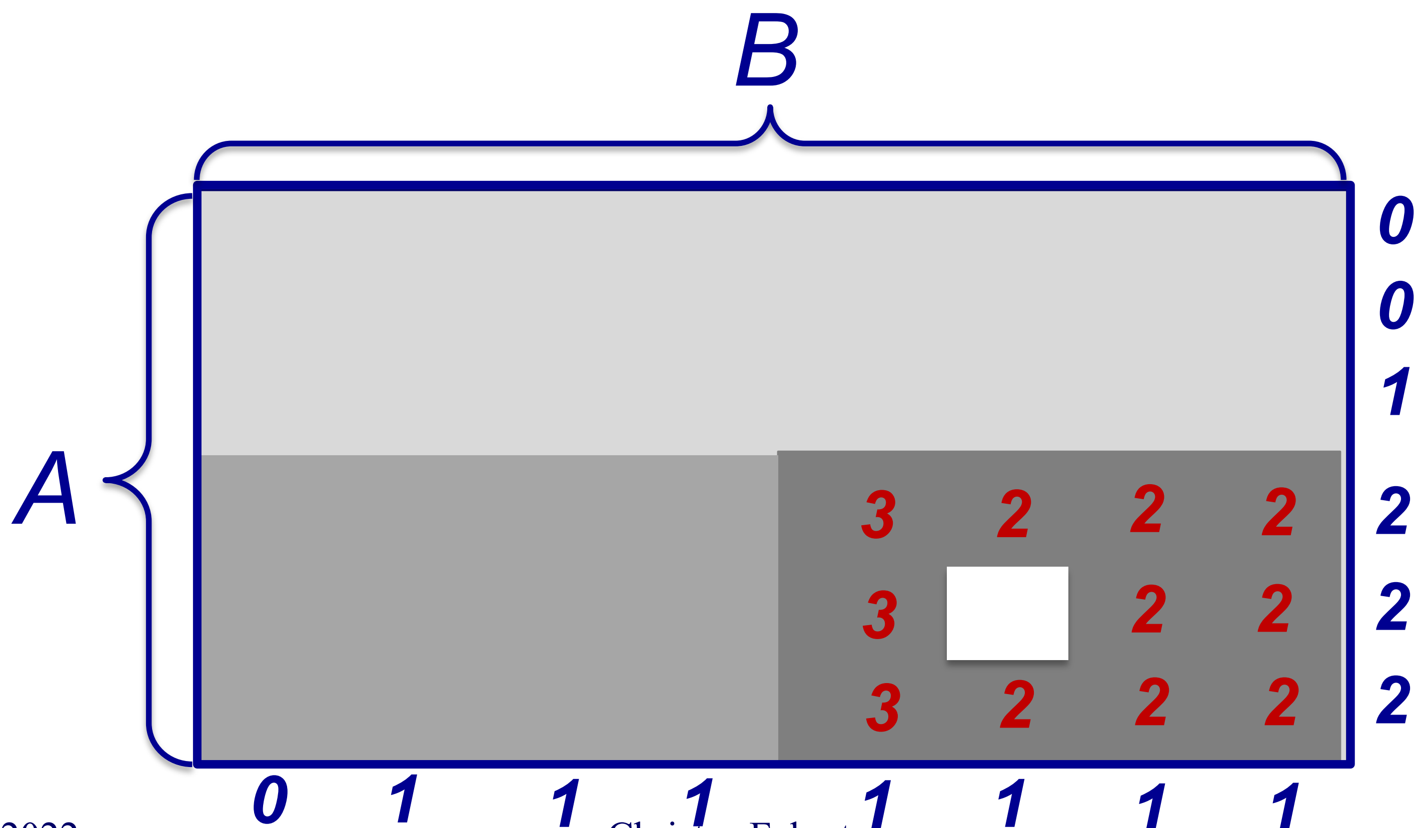
$g(A,B)$
$= (\mathbf{12}+23)/7$
$= 35/7$

Carnegie Mellon

Details

Greedy Algorithm

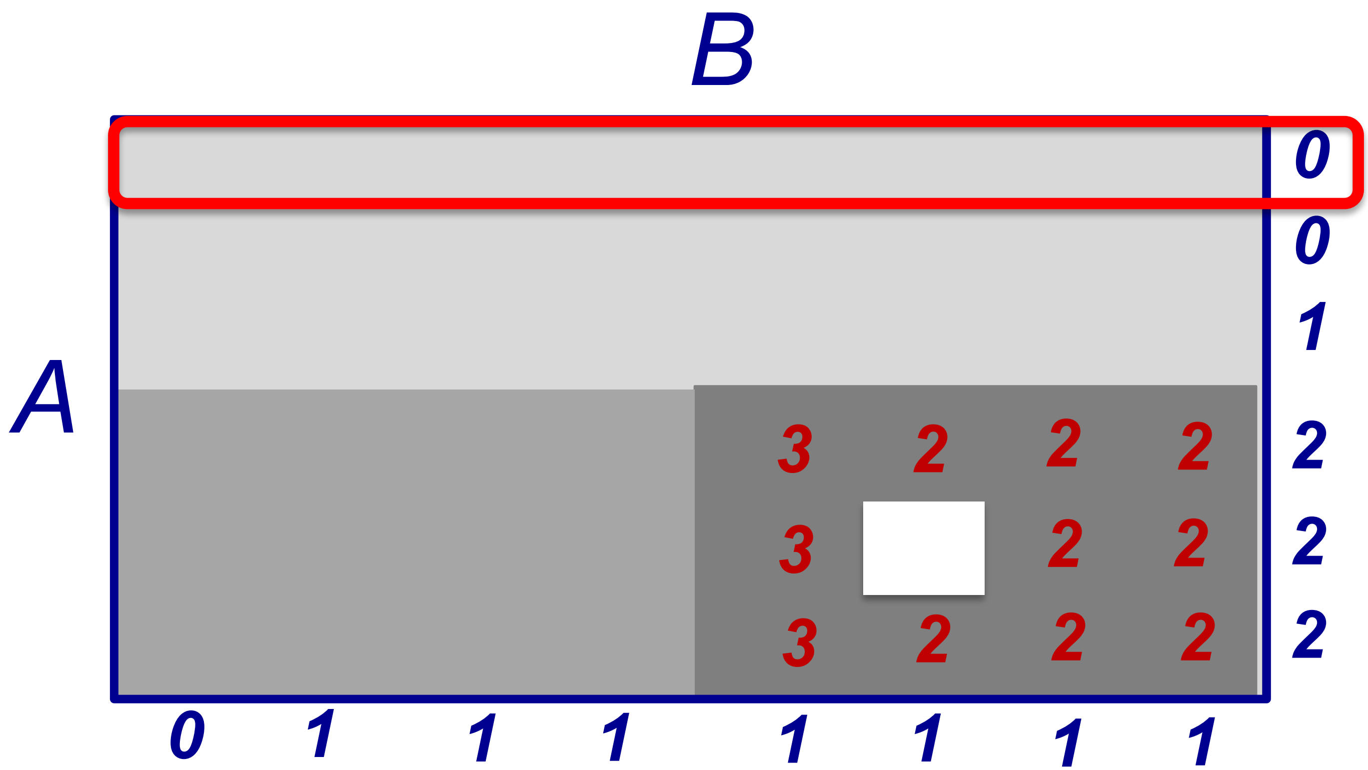
Start with sets A, B as all users / products



Details

Greedy Algorithm

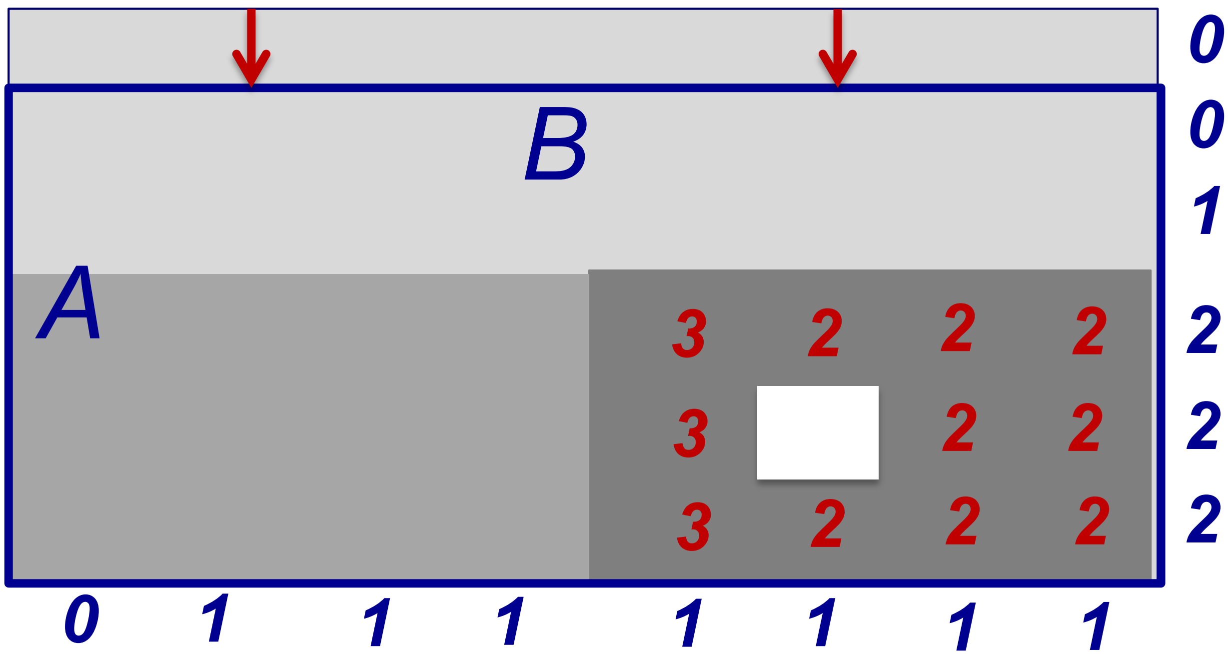
Delete rows / columns greedily to maximize g (average suspiciousness)



Details

Greedy Algorithm

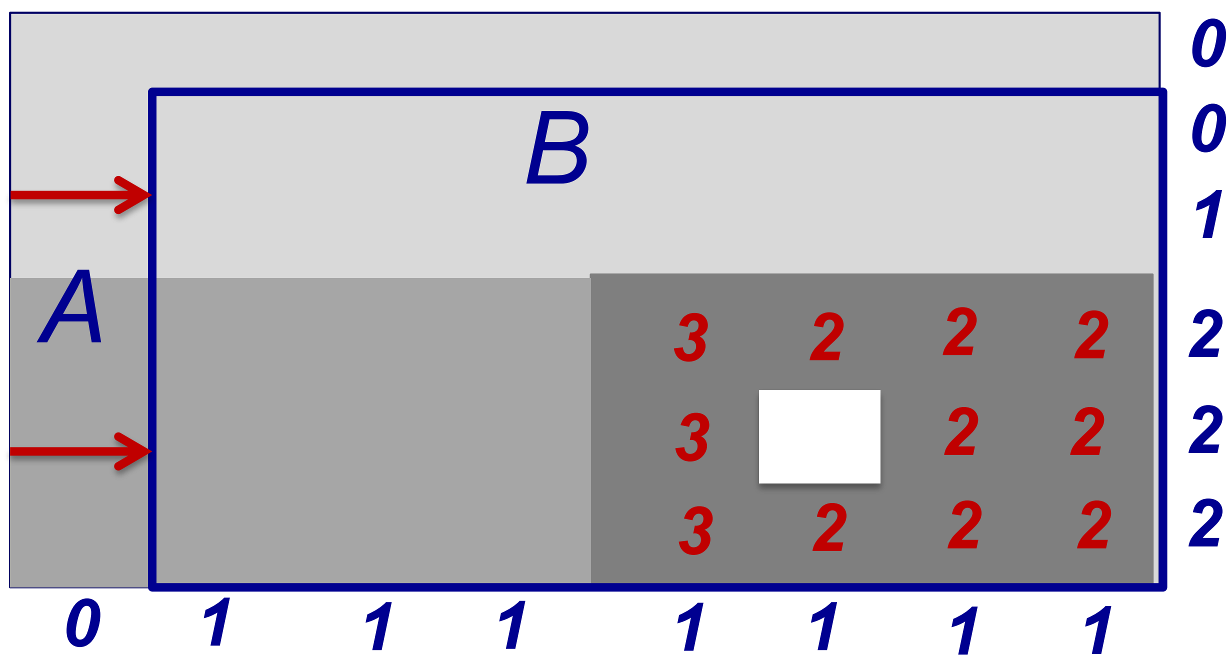
Delete rows / columns greedily to maximize g (average suspiciousness)



Details

Greedy Algorithm

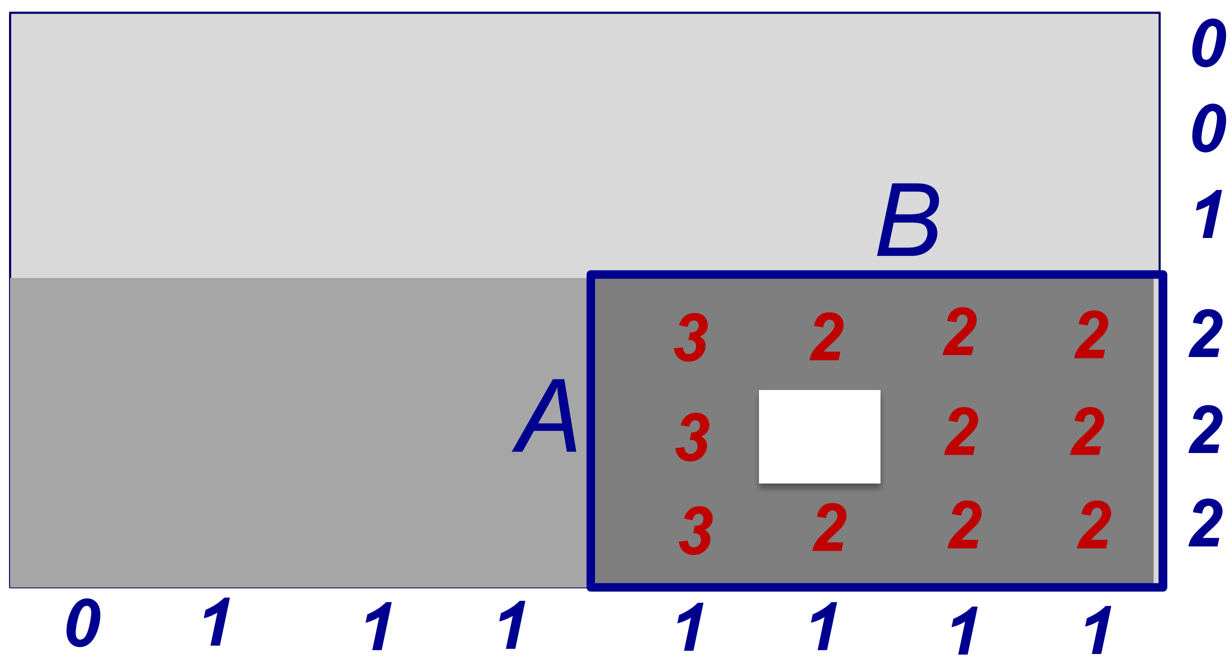
Delete rows / columns greedily to maximize g (average suspiciousness)



Details

Greedy Algorithm

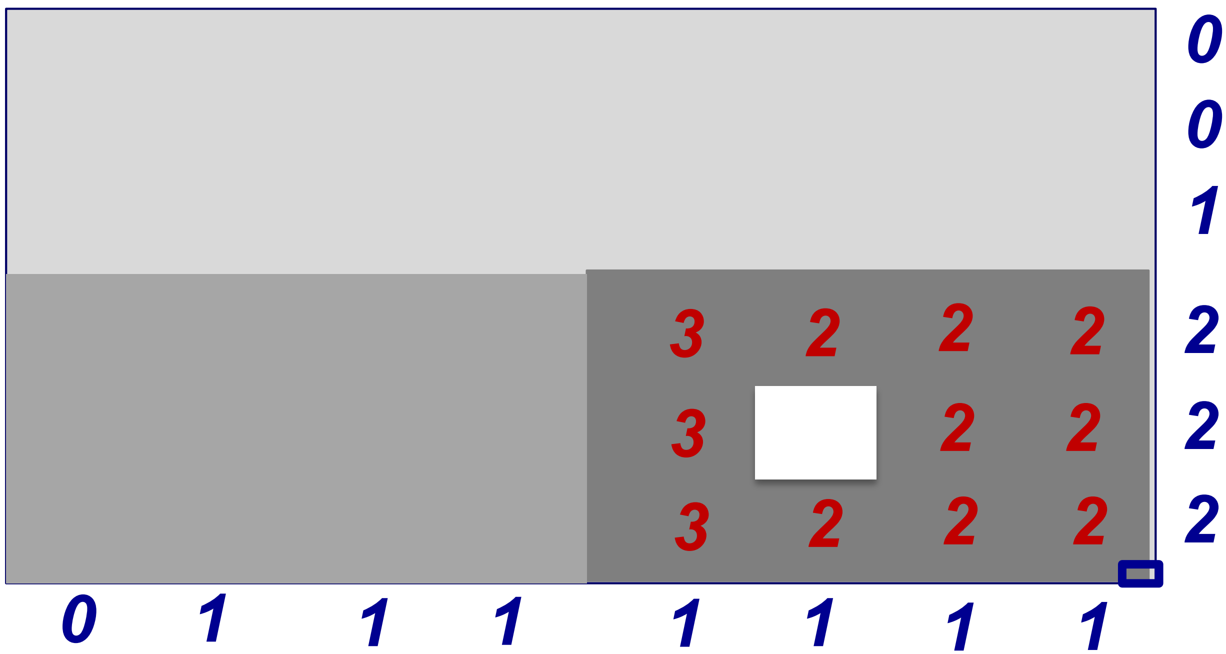
Delete rows / columns greedily to maximize g (average suspiciousness)



Details

Greedy Algorithm

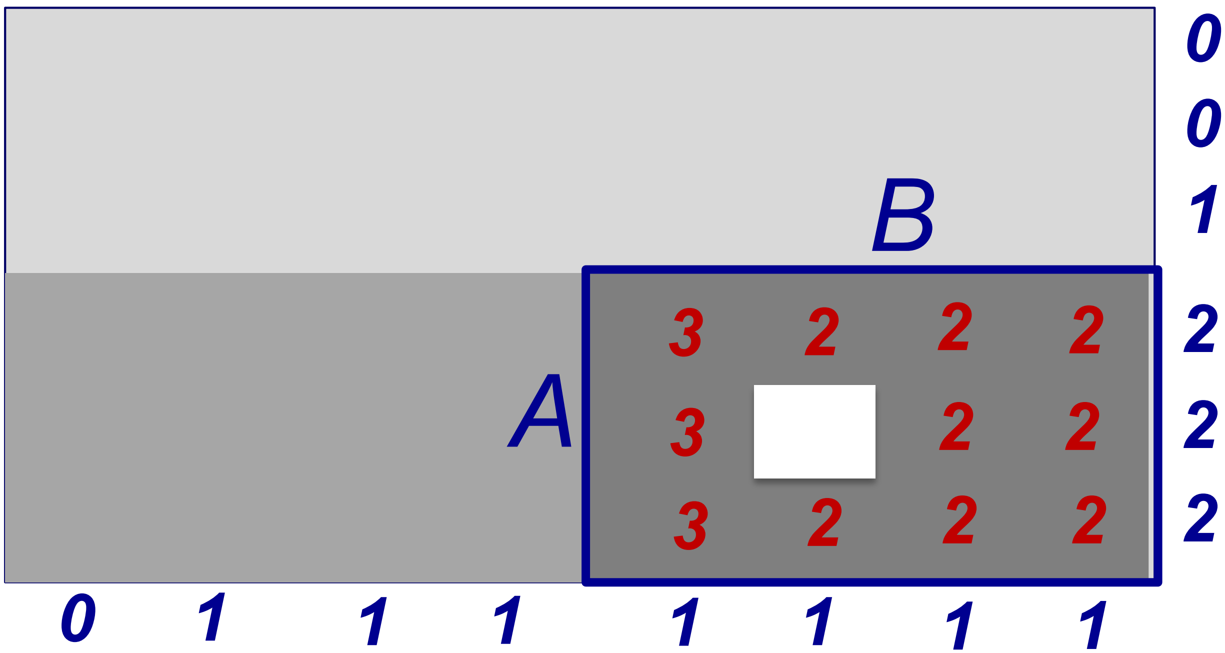
Continue until A and B are empty



Details

Greedy Algorithm

Return the best subsets A and B seen so far (based on g)



Details

Theoretical guarantee

Thm 1: The subgraph (A,B) returned by FRAUDAR satisfies

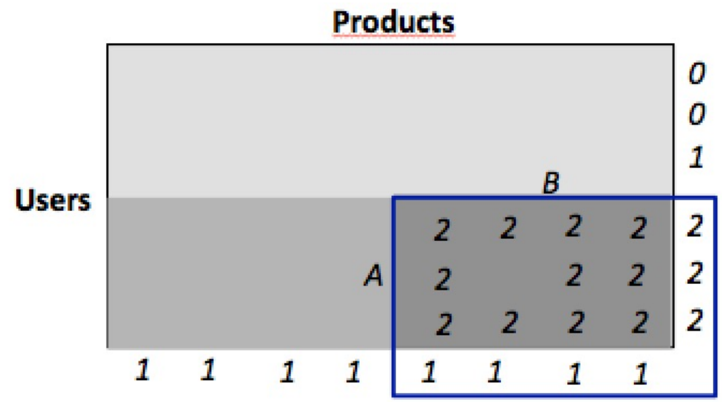
$$\boxed{g(\mathcal{A} \cup \mathcal{B}) \geq \frac{1}{2} g_{OPT}}$$

FRAUDAR subgraph

Optimum value of g

FRAUDAR - Conclusion

- Suspiciousness metric

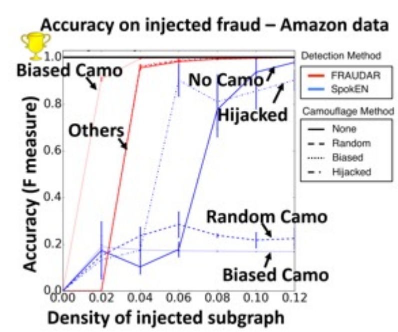


$$g(A,B) = f(A,B) / (|A| + |B|)$$

- Theoretical guarantees

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

- Effectiveness

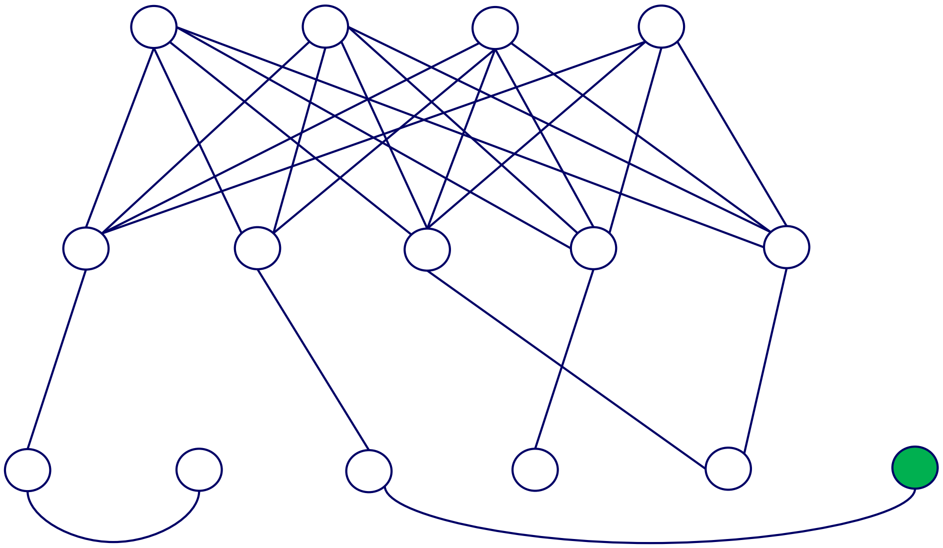


Roadmap

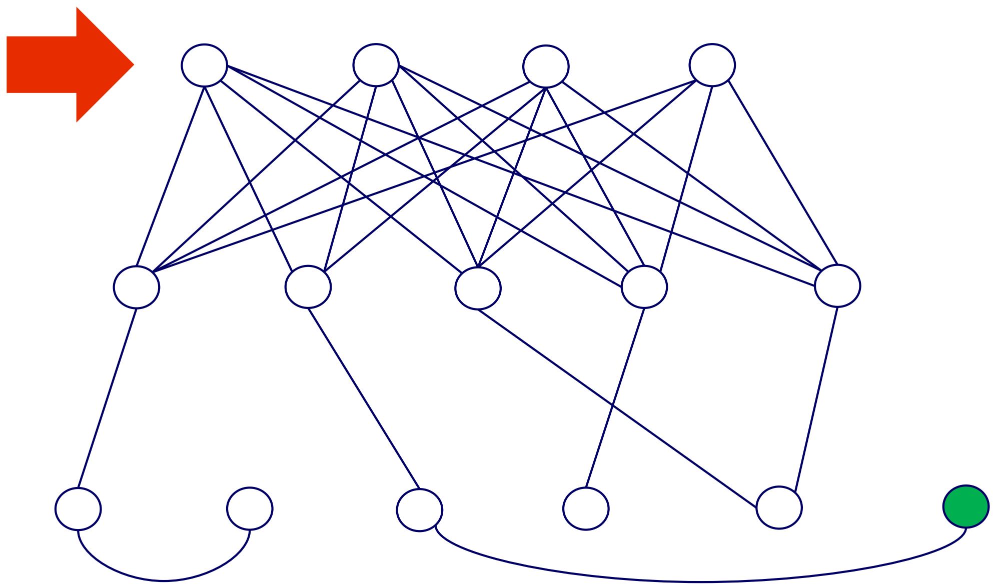
- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral methods
    - No labels – dense subgraphs
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

E-bay Fraud detection

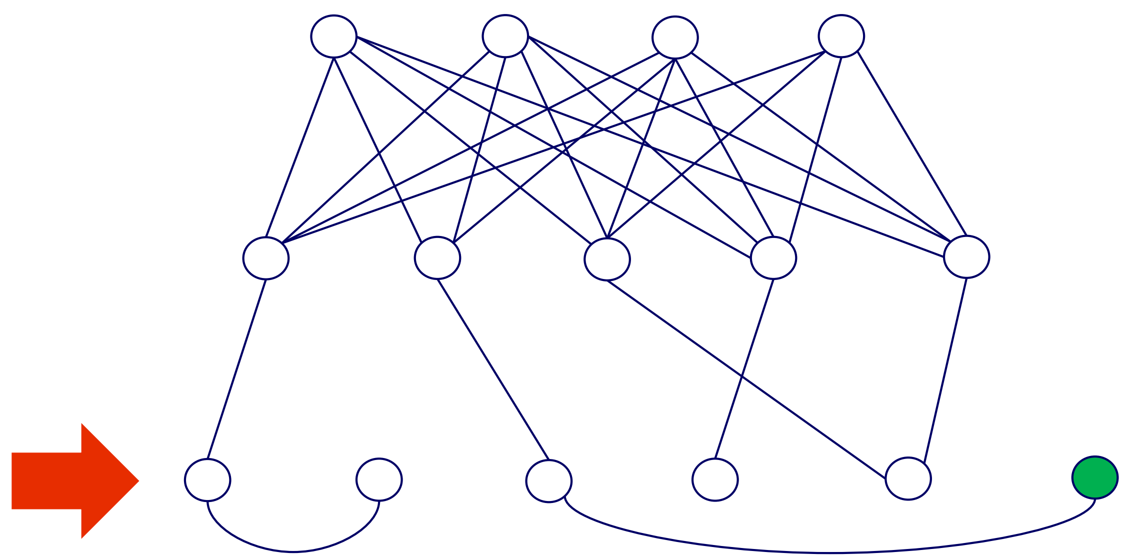
w/ Polo Chau &
Shashank Pandit, CMU
[www'07]



E-bay Fraud detection

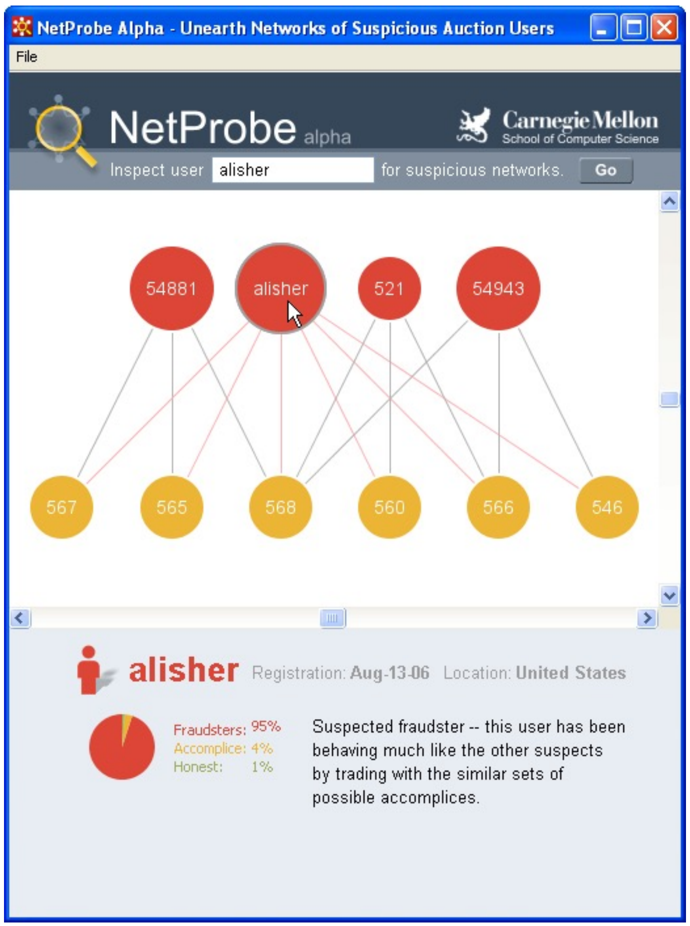


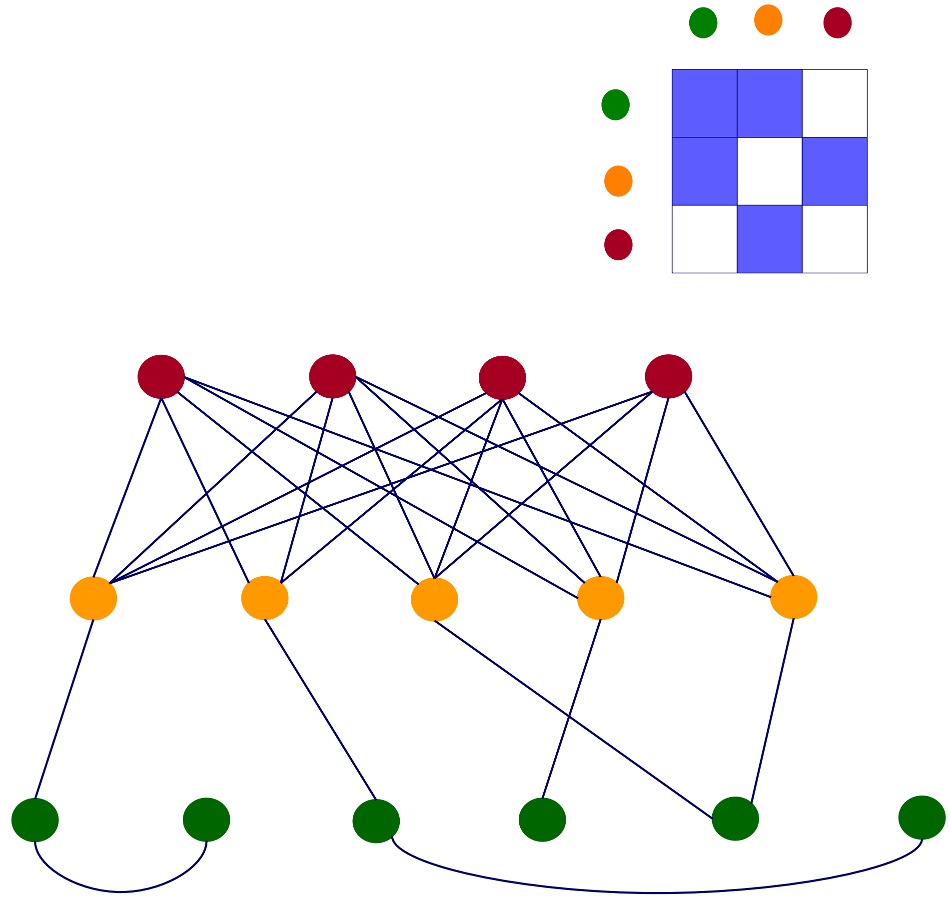
E-bay Fraud detection



Carnegie Mellon

E-bay Fraud detection - NetProbe





Popular press

USA TODAY

The Washington Post

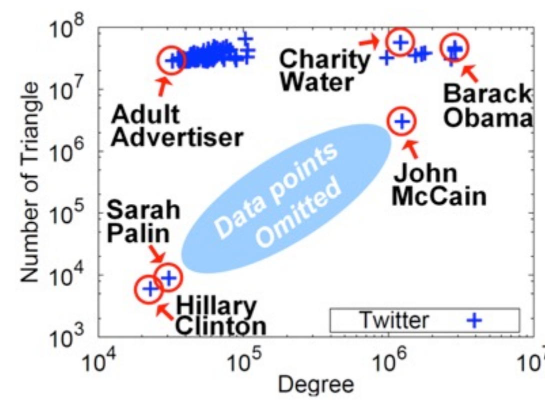
Los Angeles Times

And less desirable attention:

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

Summary of Part#1

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



Patterns  anomalies

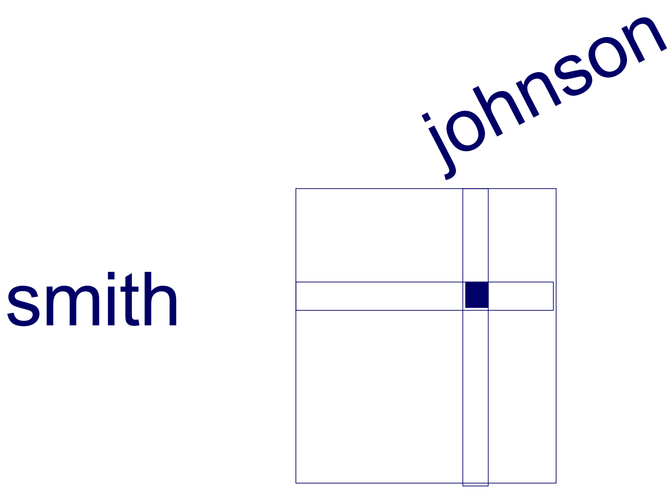
Roadmap

- 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

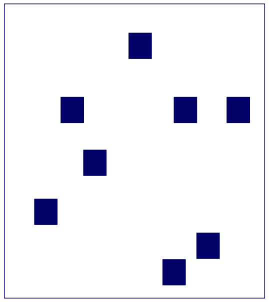
Graphs over time -> tensors!

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



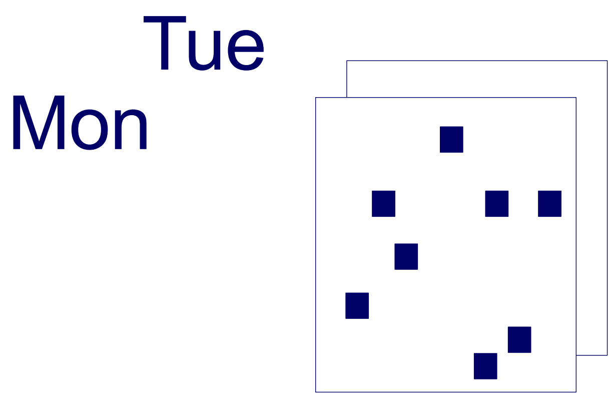
Graphs over time -> tensors!

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



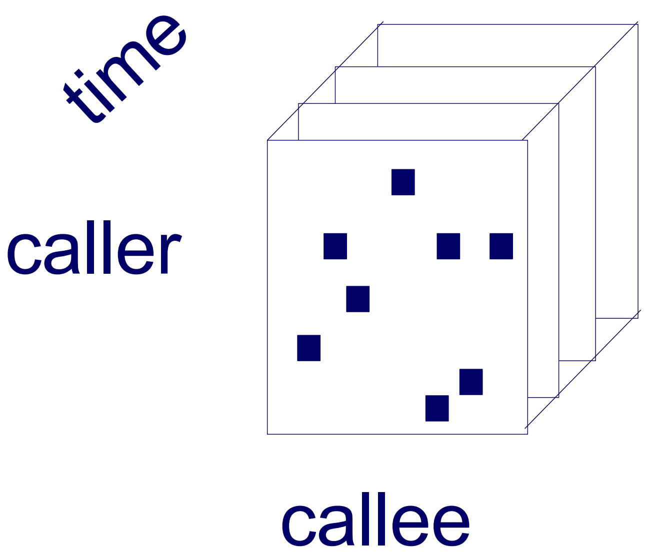
Graphs over time -> tensors!

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



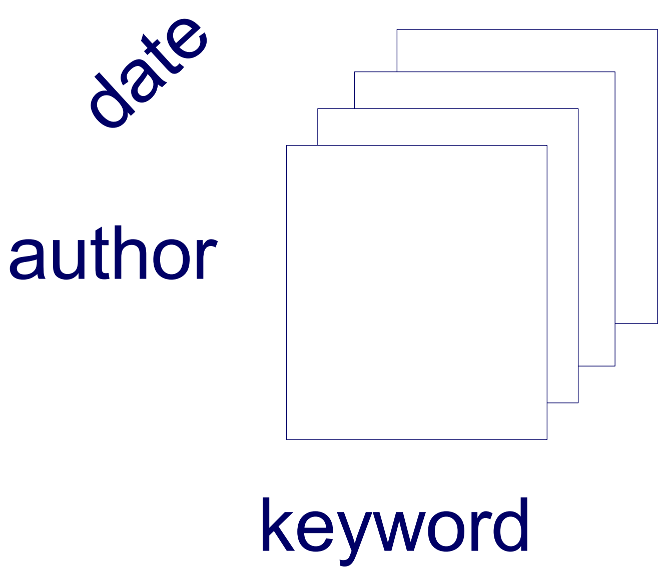
Graphs over time -> tensors!

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



Graphs over time -> tensors!

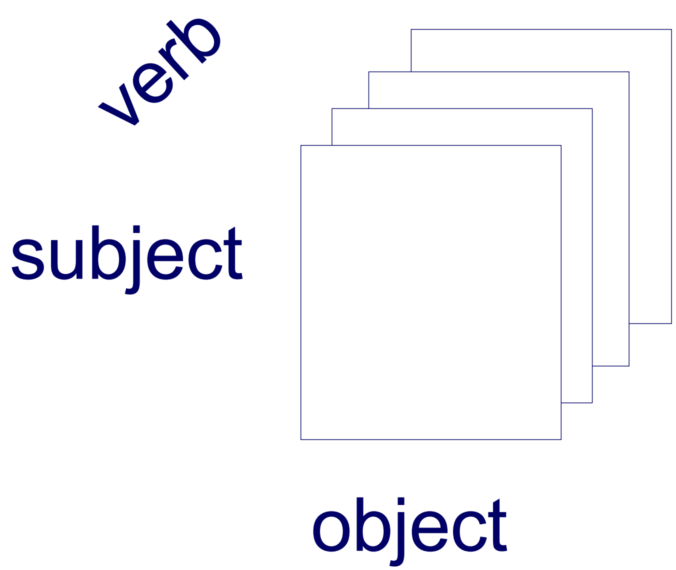
- Problem #2.1’:
  - Given author-keyword-date
  - Find patterns / anomalies



MANY more settings, with >2 ‘modes’

Graphs over time -> tensors!

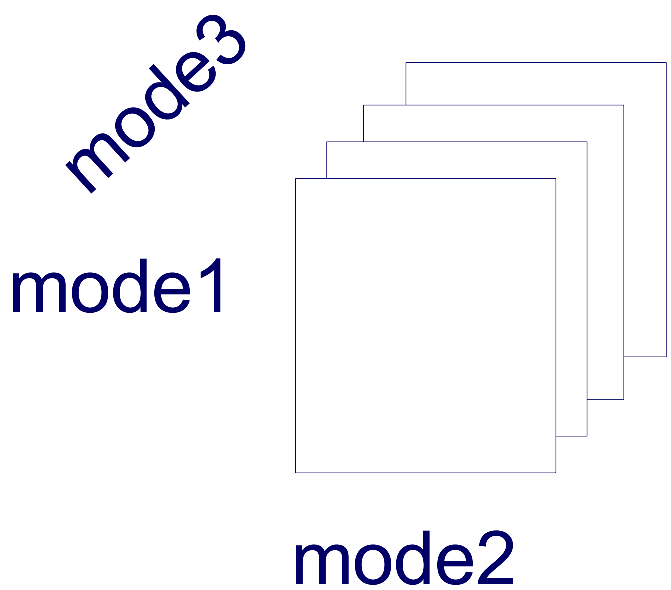
- Problem #2.1'':
  - Given subject – verb – object facts
  - Find patterns / anomalies



MANY more settings, with >2 'modes'

Graphs over time -> tensors!

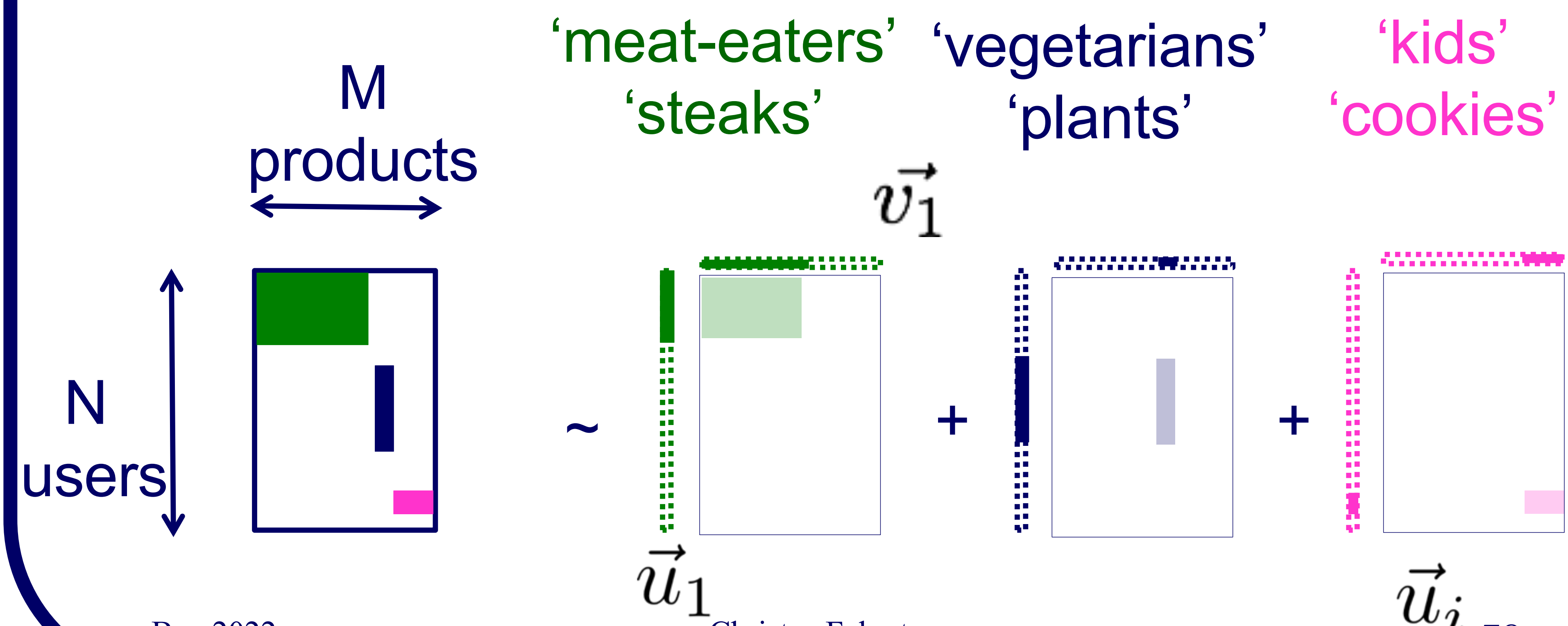
- Problem #2.1''':
  - Given <triplets>
  - Find patterns / anomalies



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

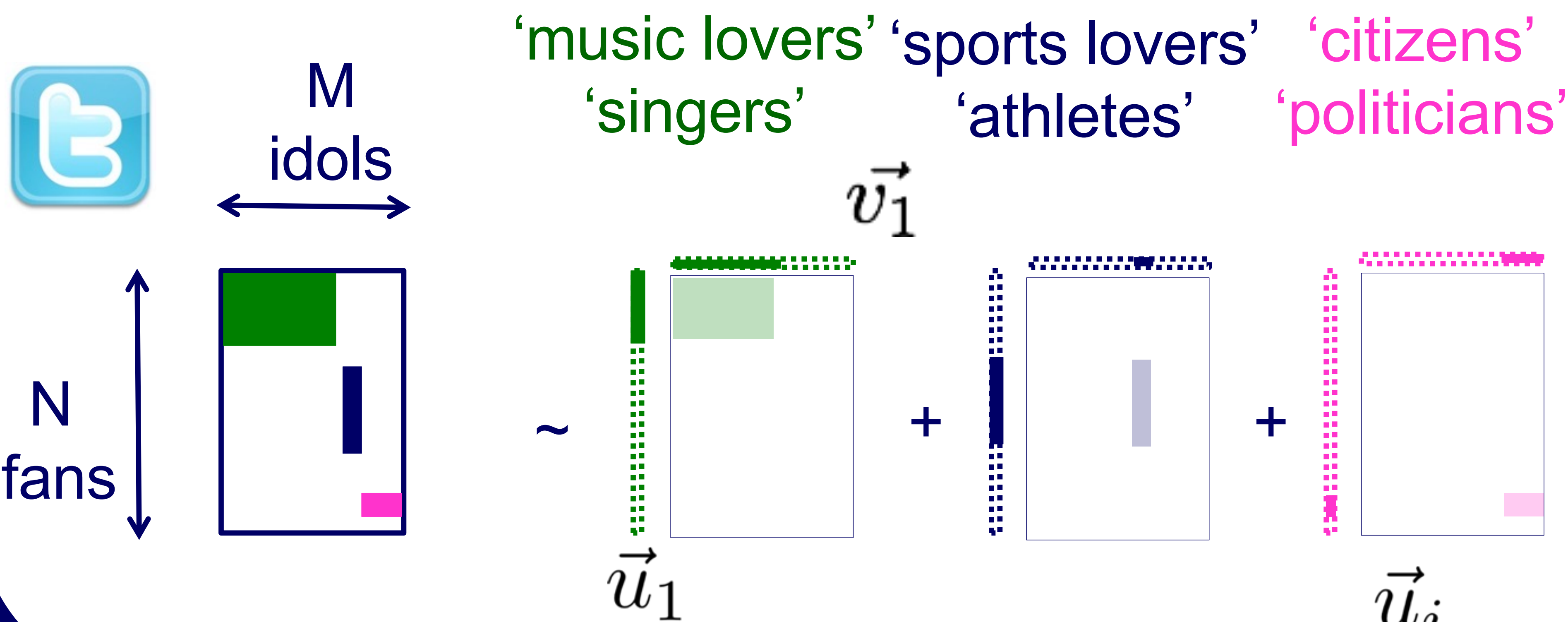
Answer : tensor factorization

- Recall: (SVD) matrix factorization: finds blocks



Crush intro to SVD

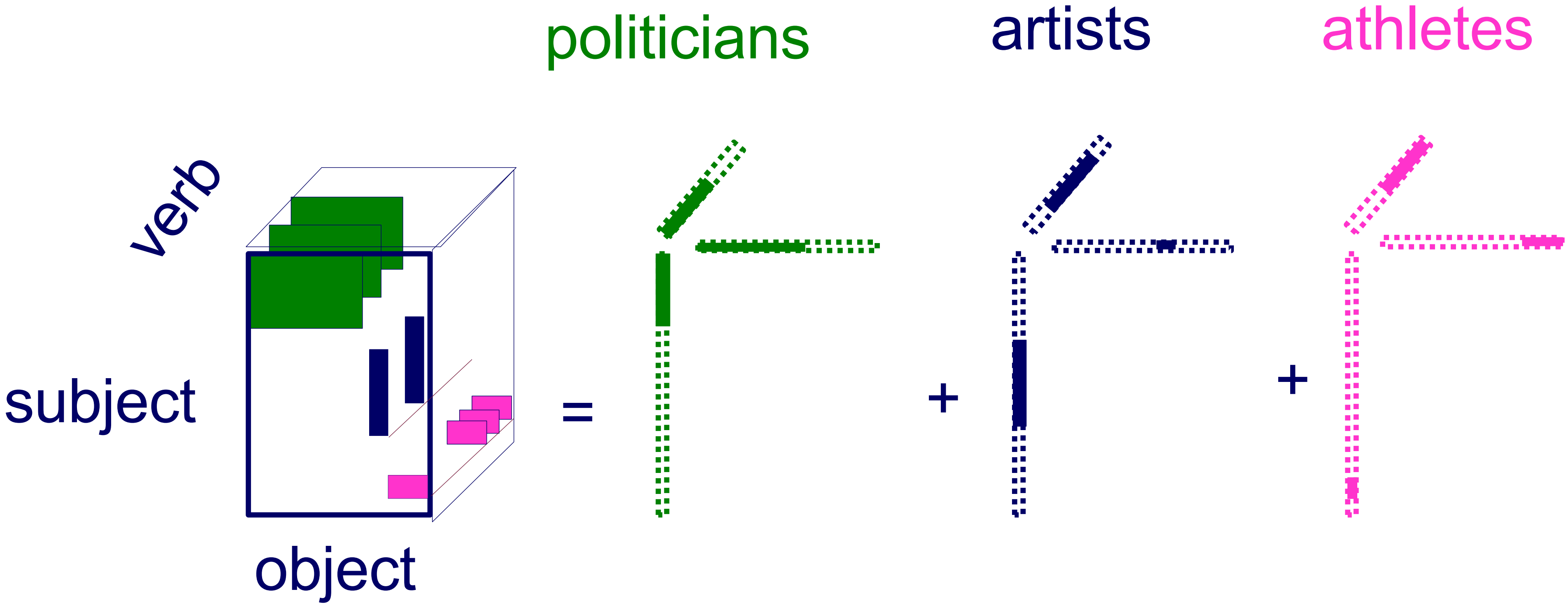
- Recall: (SVD) matrix factorization: finds blocks



Carnegie Mellon

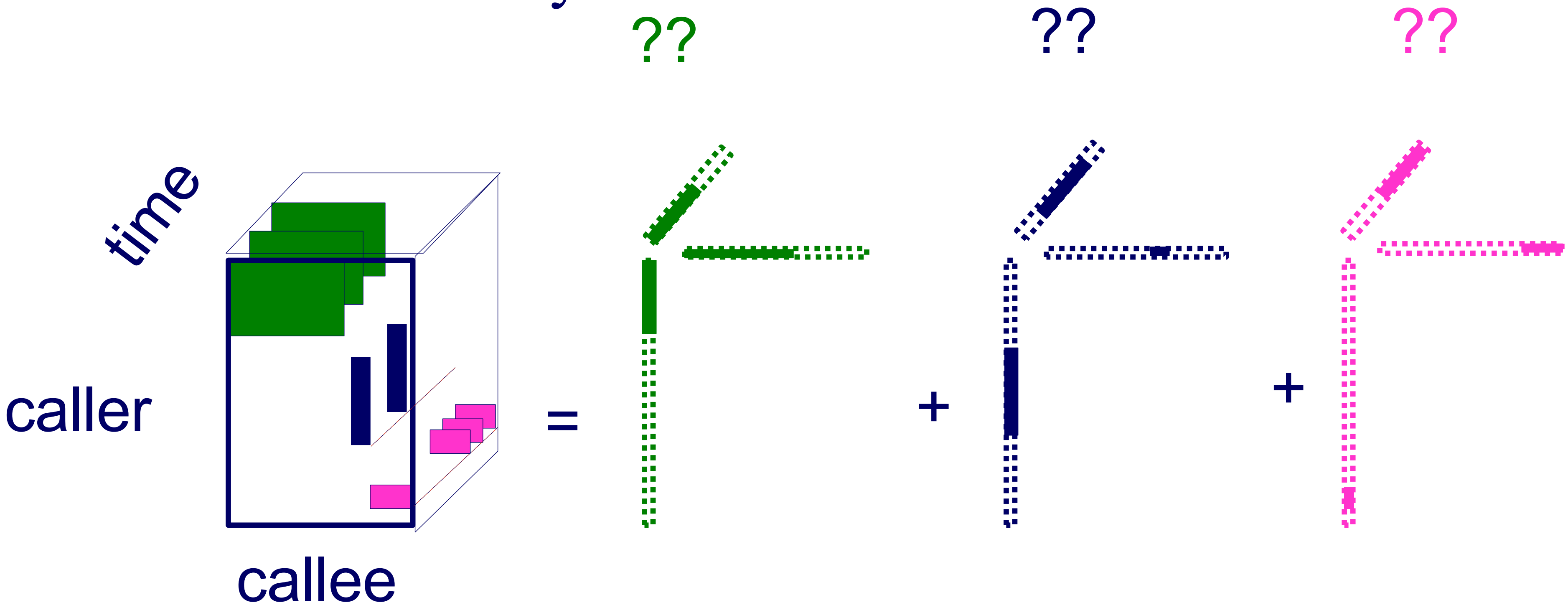
Answer: tensor factorization

- PARAFAC decomposition



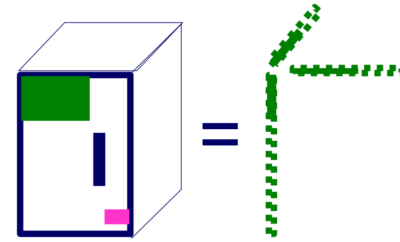
Answer: tensor factorization

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



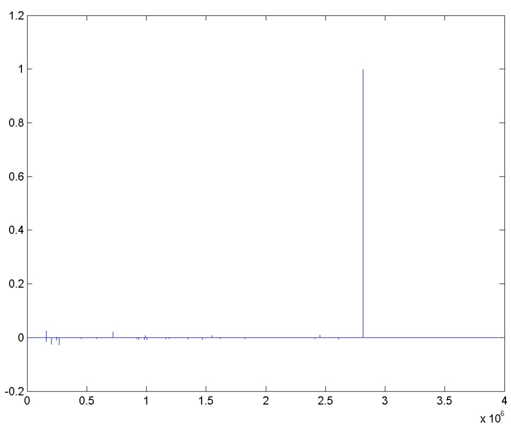
Carnegie Mellon

Anomaly detection in time-evolving graphs

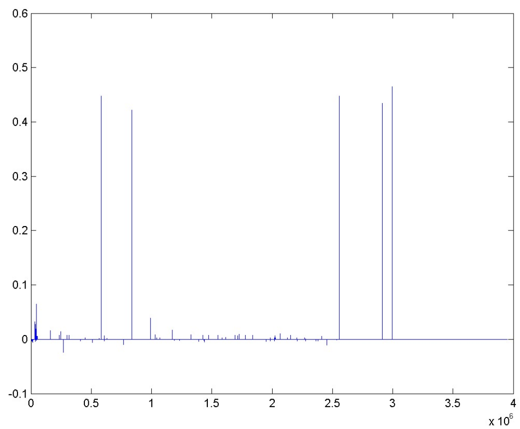


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

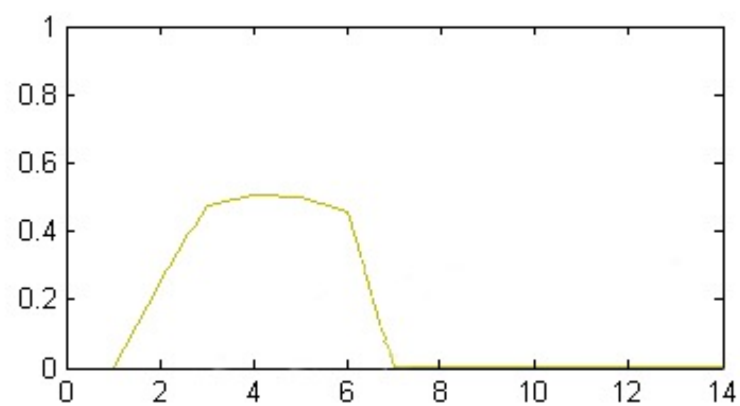
1 caller



5 receivers



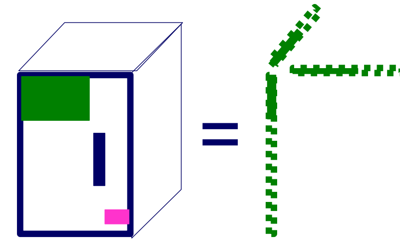
4 days of activity



~200 calls to EACH receiver on EACH day!

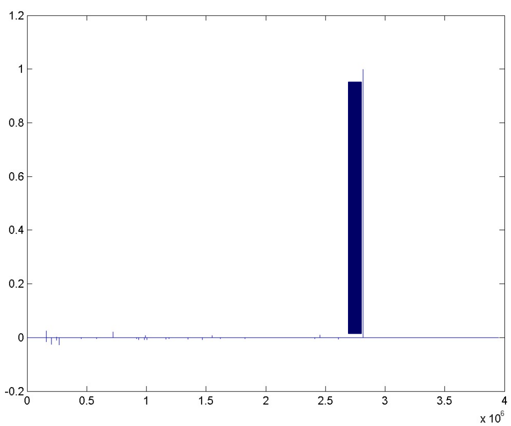
Carnegie Mellon

Anomaly detection in time-evolving graphs

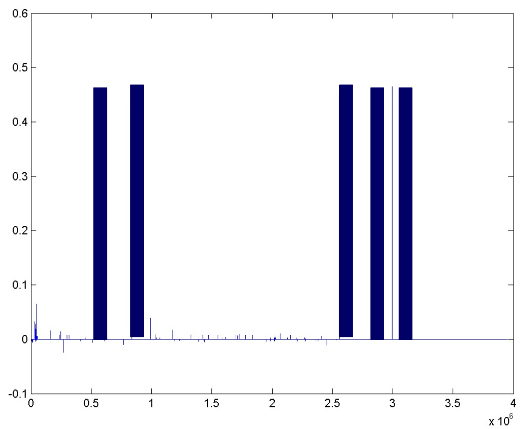


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

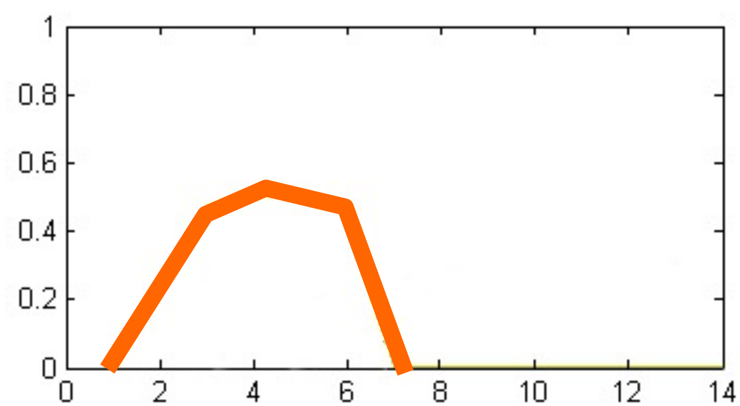
1 caller



5 receivers



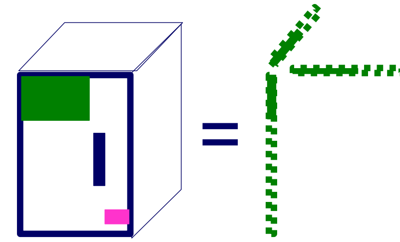
4 days of activity



~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs

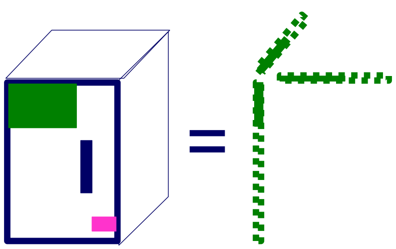


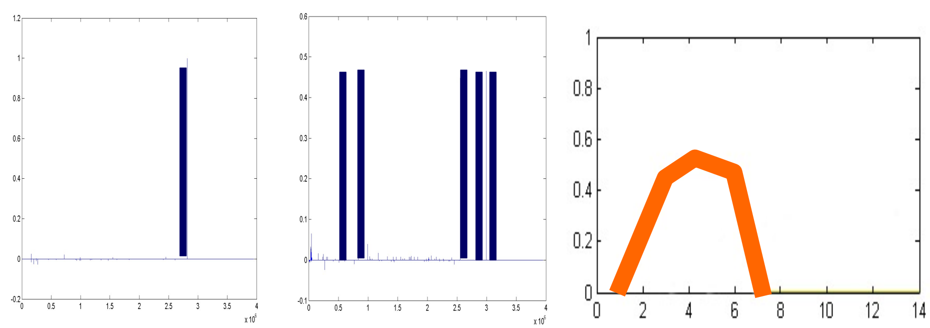
- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

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





Carnegie Mellon

Roadmap

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

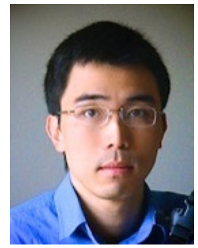
Carnegie Mellon

Cast

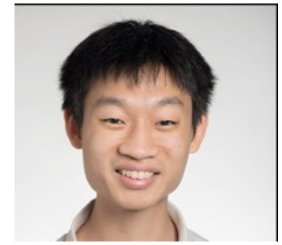

Akoglu, Leman


Araujo, Miguel

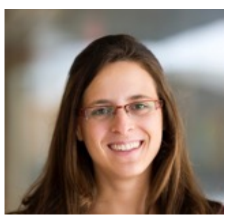

Beutel, Alex


Chau, Polo


Eswaran, Dhivya


Hooi, Bryan


Kang, U


Koutra, Danai


Papalexakis, Vagelis

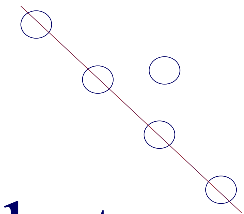

Shah, Neil


Shin, Kijung

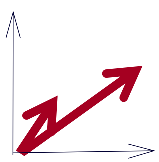

Song, Hyun Ah

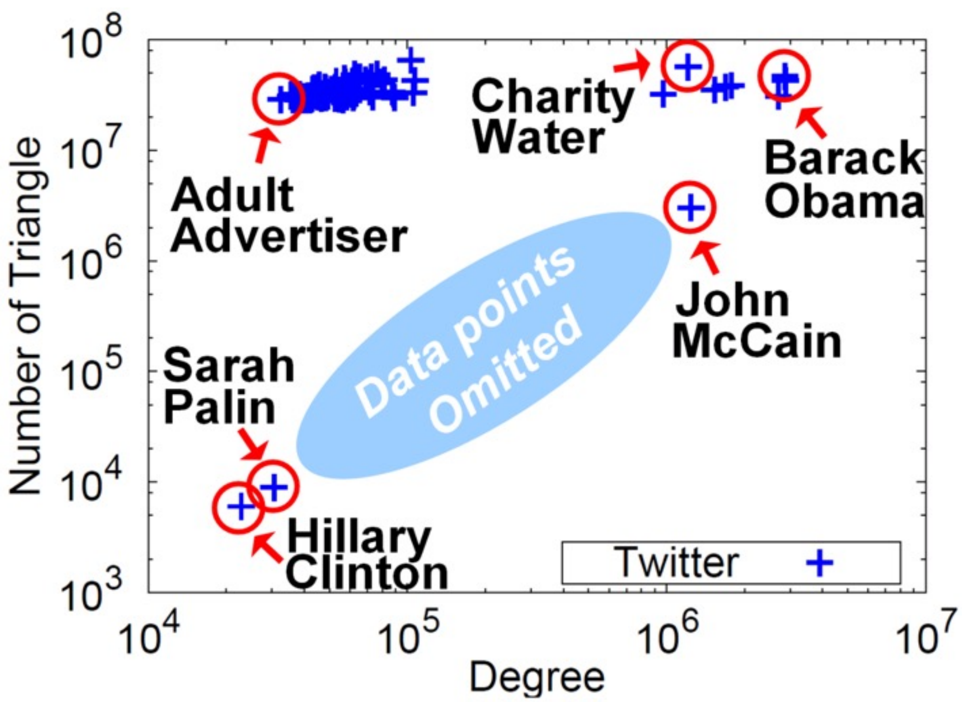
CONCLUSION#1 – Big data

- **Patterns**  **Anomalies**



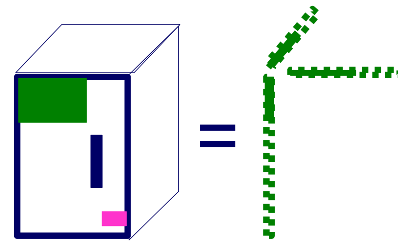
- **Large** datasets reveal patterns/outliers that are invisible otherwise

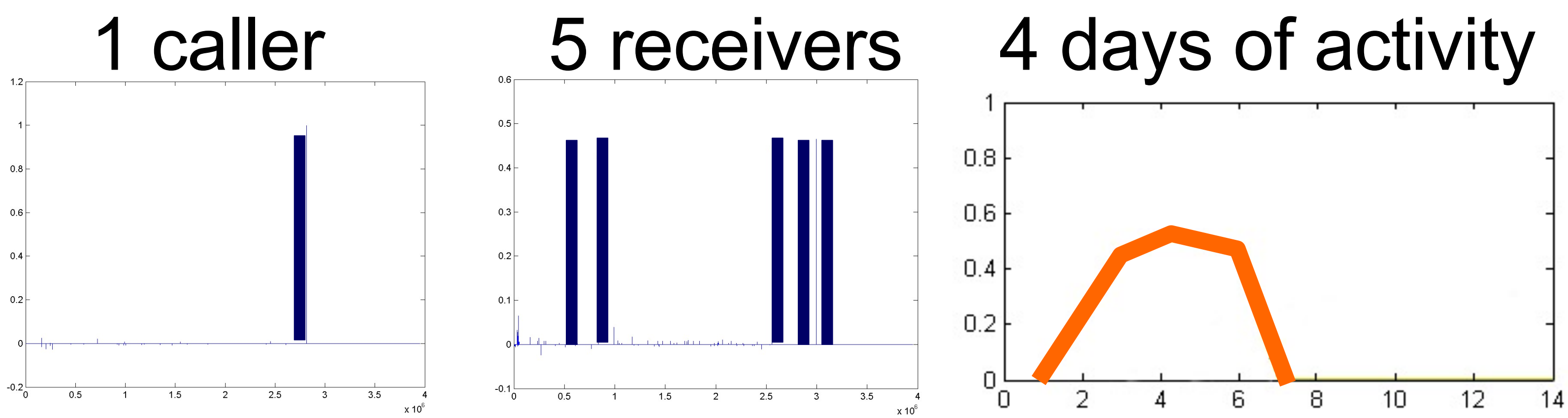




CONCLUSION#2 – tensors

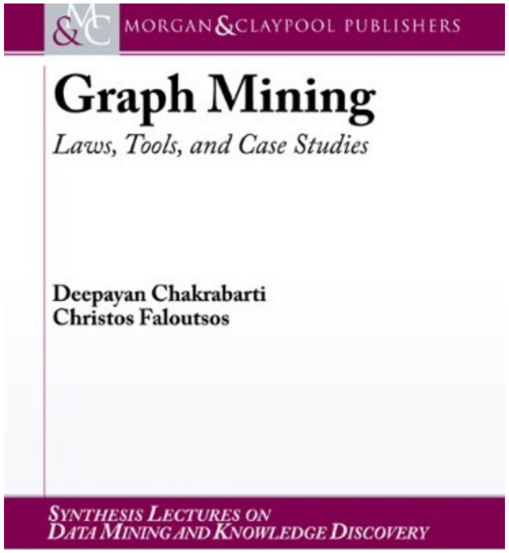
- powerful tool





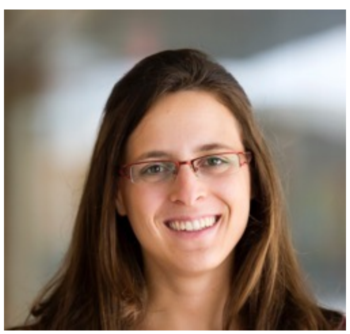
References

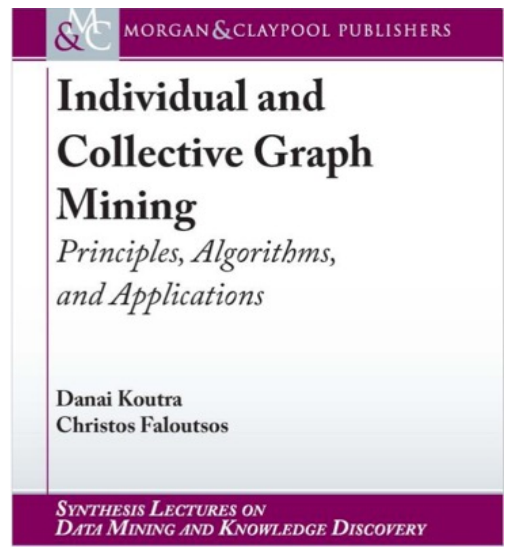
- D. Chakrabarti, C. Faloutsos: *Graph Mining – Laws, Tools and Case Studies*, Morgan Claypool 2012
- http://www.morganclaypool.com/doi/abs/10.2200/S00449ED1V01Y201209DMK006



References

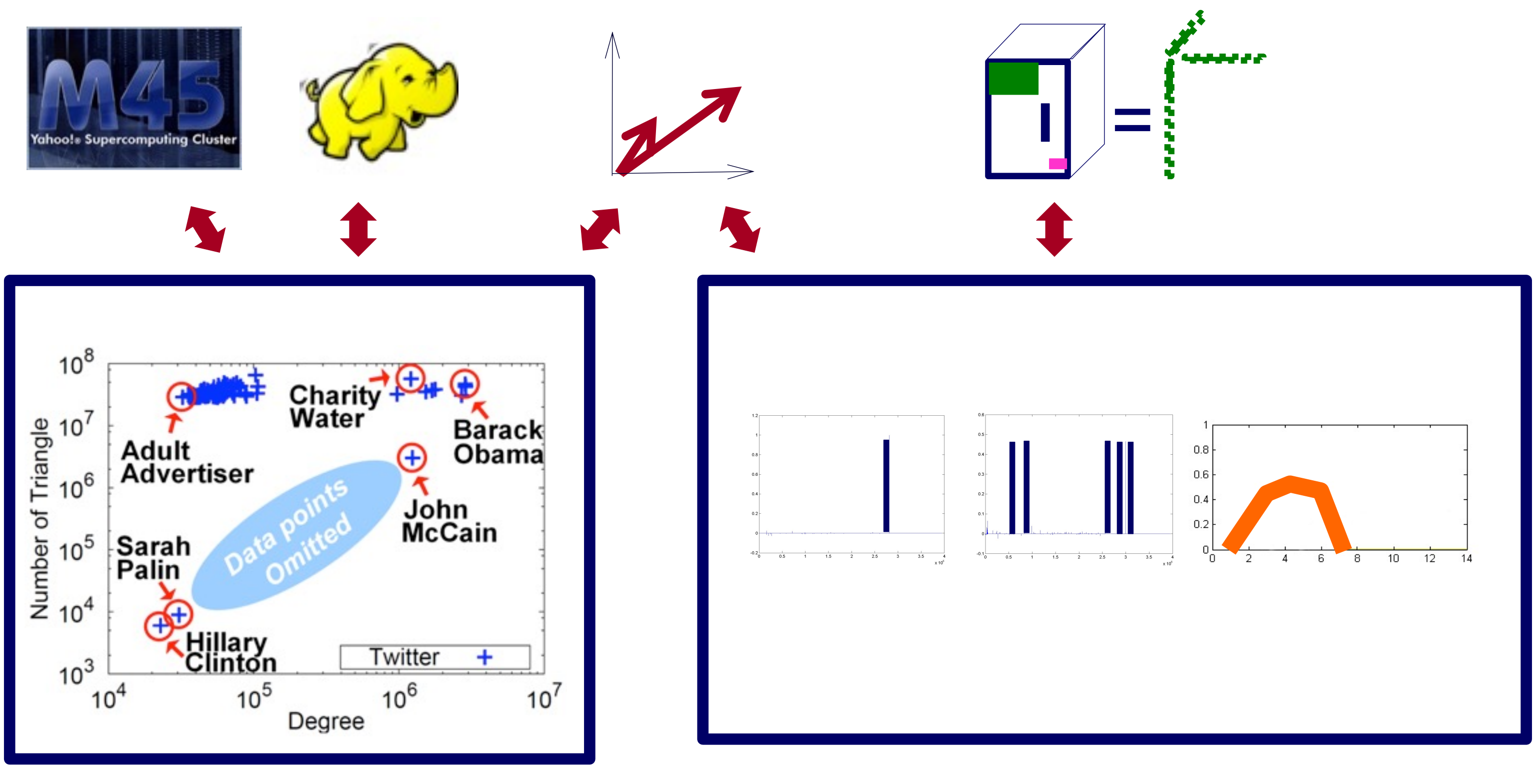
- Danai Koutra and Christos Faloutsos, *Individual and Collective Graph Mining: Principles, Algorithms, and Applications, Morgan Claypool 2017* (https://doi.org/10.2200/S00796ED1V01Y201708DMK014)





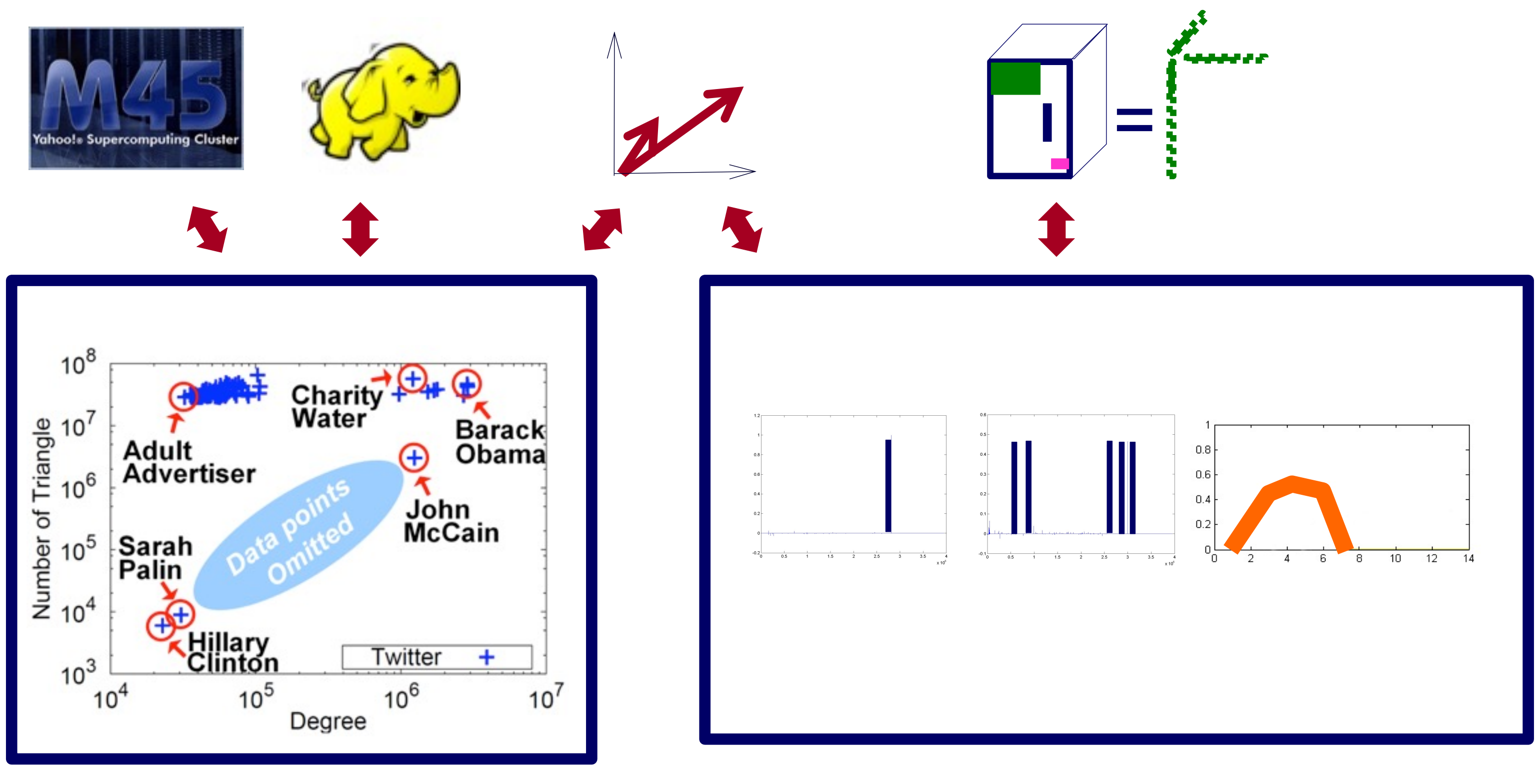
TAKE HOME MESSAGE:

Cross-disciplinarity



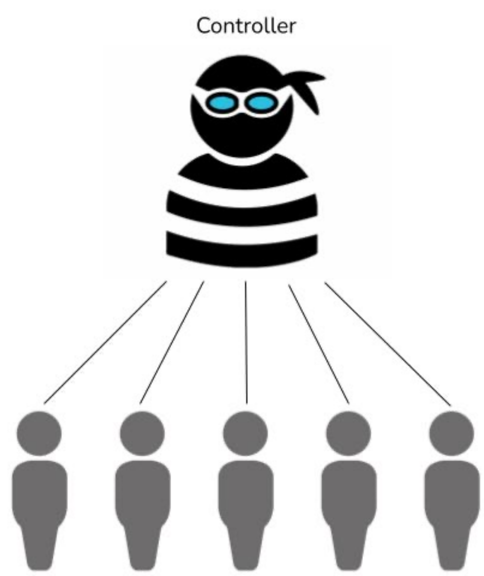
Thank you!

Cross-disciplinarity



Bonus material

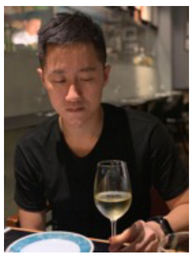
- Human trafficking detection
- == near-duplicate document detection



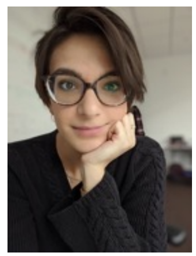
Carnegie Mellon

InfoShield: Generalizable Information-Theoretic Human Trafficking Detection

ICDE 2021

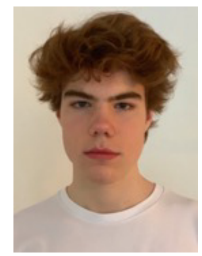


Meng-Chieh Lee*



Catalina Vajiac*

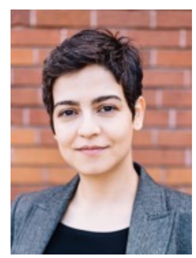
Aayushi Kulshrestha



Sacha Levy

Namyong Park

Cara Jones

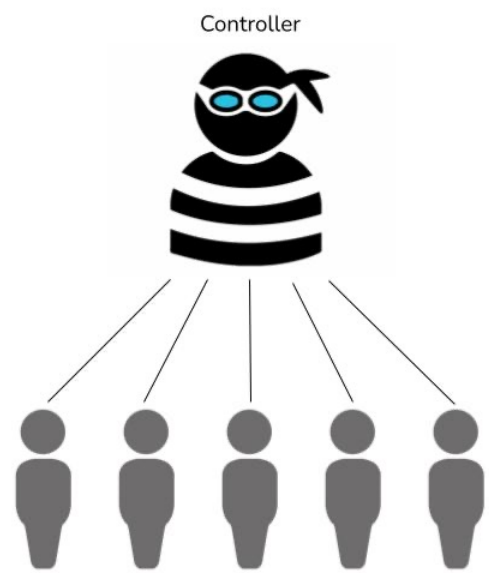


Reihaneh Rabbany

Christos Faloutsos

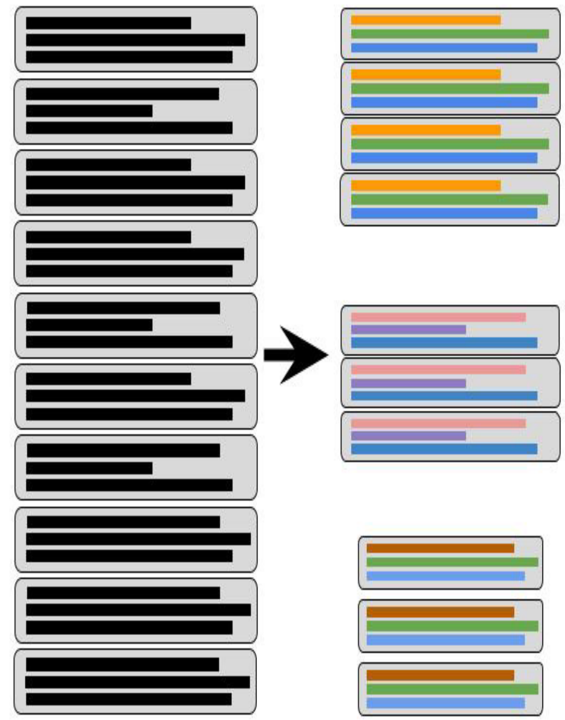
Motivation

- *Pervasive:* Millions exploited every year
- *Room for Improvement:* law enforcement looks at ads manually
- *How can we separate* HT ads from the rest?



Problem definition:

- *Insight:* controllers write ads for all their victims, which makes the text similar.
- What can we do?
  - – - *Group* ads into micro-clusters based on text
  - – - *Visualize* each



Carnegie Mellon

Toy example

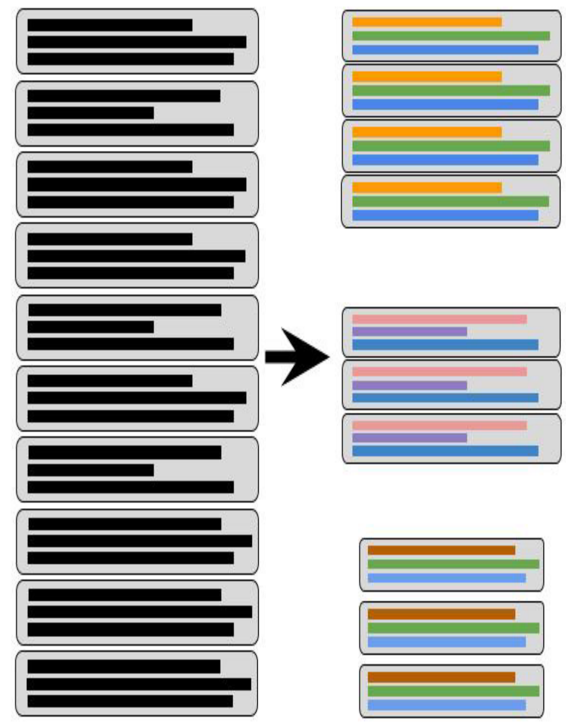
Constant Slot Insertion Deletion Substitution

T_1	**This is a great**	*****	**and the**	*****	**dollar price is**		**great**
#1	This is a great	soap,	and the	5	dollar price is		great
#2	This is a great	chair,	and the	10	dollar price is		great
#3	This is a great	hat,	and the	3	dollar price is		great
#4	This is great	blue pen,	and the	3	dollar price is	so	good

T_2	**I made 30k working**	*****	**- call**	*****	**or visit**	*****
#5	I made 30k working	on this job	- call	123-456.7890	or visit	scam.com
#6	I made 30k working	from home	- call	123-456.7890	or visit	fraud.com

Problem definition:

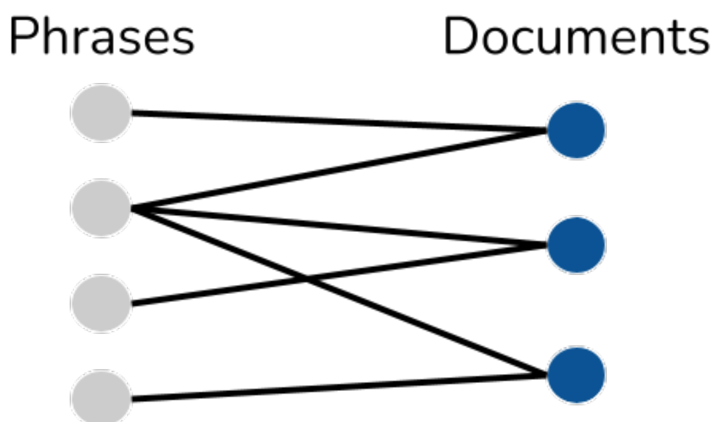
How?



Details

Infoshield-Coarse: overview

- 1. Given a document: *Extract tf-idf scores* for each phrase
- 2. *Create a bipartite graph* of documents and top 10% of phrases
- 3. Once all documents are processed, *return connected components*



Details

Infoshield-Fine

- Group similar documents
- To minimize ‘Description length’ (MDL)

Constant Slot Insertion Deletion Substitution

T_1		**sismo**	**richter**		**km al sureste de puerto escondido oax lat lon pf km**
#1		sismo	richter		km al sureste de puerto escondido oax lat lon pf km
Omit	21 Identical	Tweets as #1 ...			
#23	sismologicomx	sismo	magnitud	loc	km al sureste de puerto escondido oax lat lon pf km

Results: Interpretability

Constant Slot Insertion Deletion Substitution

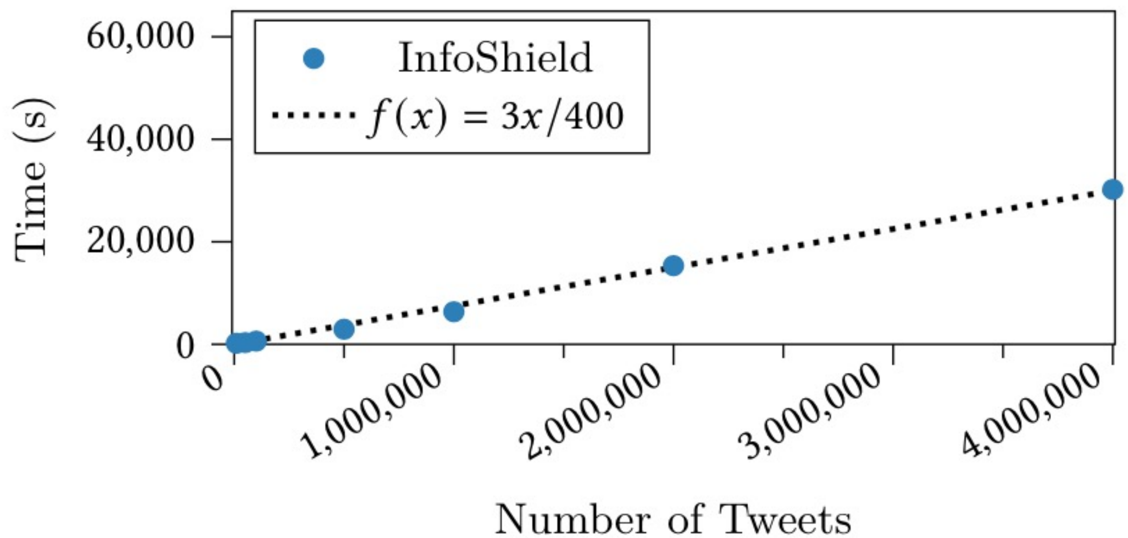
T_1		**sismo**	**richter**		**km al sureste de puerto escondido oax lat lon pf km**
#1		sismo	richter		km al sureste de puerto escondido oax lat lon pf km
Omit	21 Identical	Tweets as #1 ...			
#23	sismologicomx	sismo	magnitud	loc	km al sureste de puerto escondido oax lat lon pf km

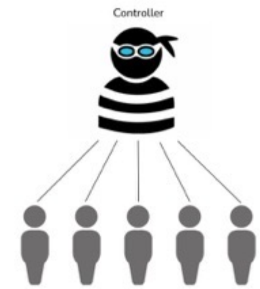
Results: Interpretability

Constant Slot Insertion Deletion Substitution

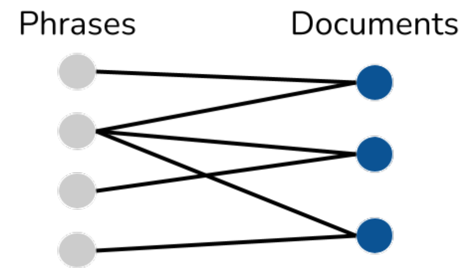
T_1	**not shown for victim's safety**							
#1	(empty)		time		(empty)		(empty)	
#2	personal description		time		(empty)		cost	
#3	(empty)		time		(empty)		cost	
#4	personal description		(empty)		preferences		cost	
...18 similar ads								

Results: Scalability





Conclusions



- Graph mining / anomaly detection helps here, too
- Explainability is a must
- Visualization is extremely helpful

Constant Slot Insertion Deletion Substitution

T_1	**This is a great**	*****	**and the**	*****	**dollar price is**		**great**
#1	This is a great	soap,	and the	5	dollar price is		great
#2	This is a great	chair,	and the	10	dollar price is		great
#3	This is a great	hat,	and the	3	dollar price is		great
#4	This is great	blue pen,	and the	3	dollar price is	so	good

T_2	**I made 30k working**	*****	**- call**	*****	**or visit**	*****
#5	I made 30k working	on this job	- call	123-456.7890	or visit	scam.com
#6	I made 30k working	from home	- call	123-456.7890	or visit	fraud.com