Carnegie Mellon

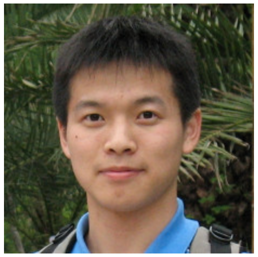
Anomaly detection in large graphs

Christos Faloutsos
CMU

Thank you!

- Annette Jiang (IEEE)

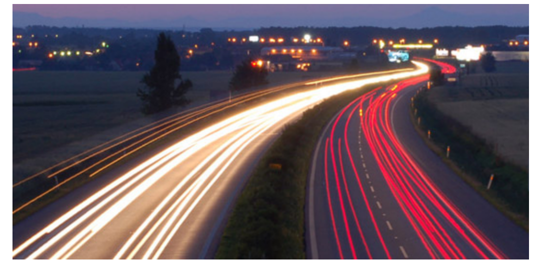
- Evan Butterfield (IEEE)



- Lei Li

Carnegie Mellon

Roadmap

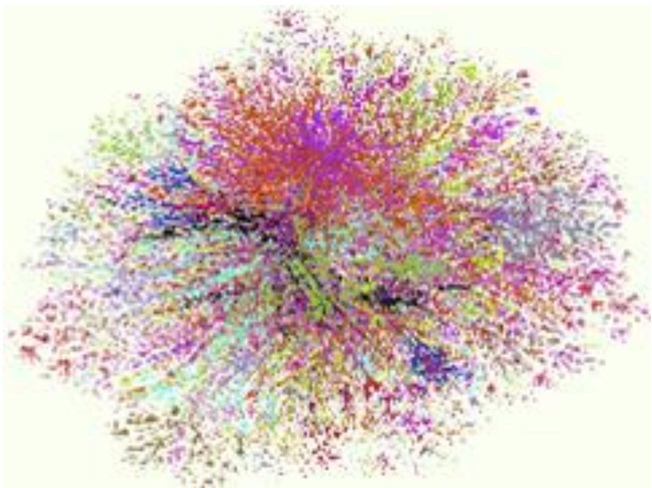


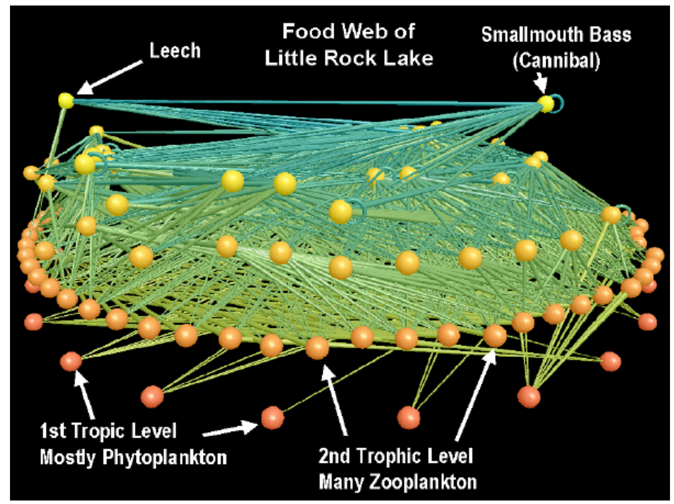
- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs; tensors
- Conclusions

Graphs - why should we care?

>$10B; ~1B users

Graphs - why should we care?





Internet Map
[lumeta.com]

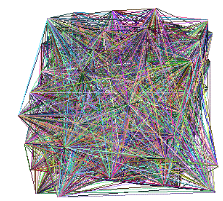
Food Web
[Martinez '91]

Graphs - why should we care?

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

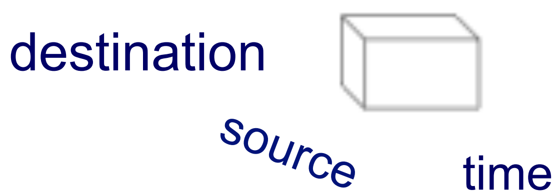
- Many-to-many db relationship -> graph

Motivating problems

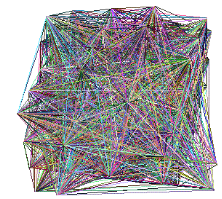


- P1: patterns? Fraud detection?

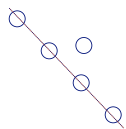
- P2: patterns in time-evolving graphs / tensors



Motivating problems

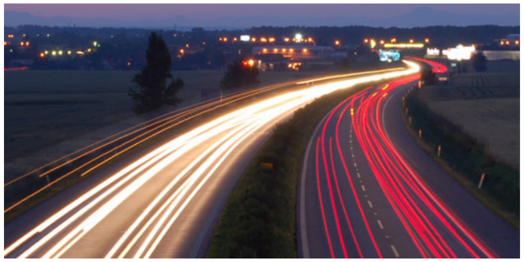


- P1: patterns? Fraud detection?

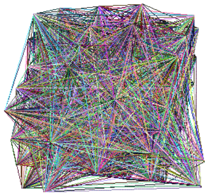
 Patterns  anomalies

- P2: patterns in time-evolving graphs / tensors

Roadmap

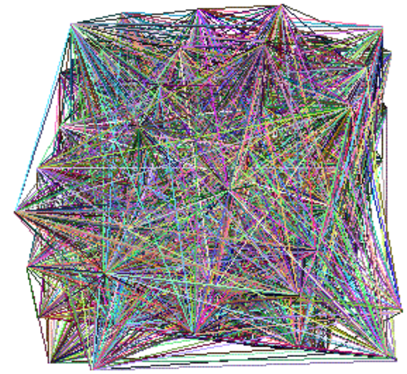


- Introduction – Motivation
  - Why study (big) graphs?
- Part#1: Patterns & fraud detection
- Part#2: time-evolving graphs; tensors
- Conclusions



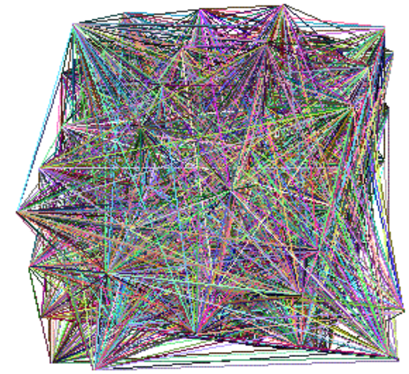
Part 1: Patterns, & fraud detection

Laws and patterns

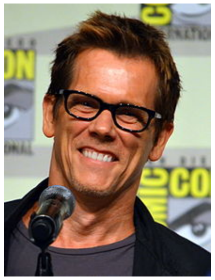


- Q1: Are real graphs random?

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

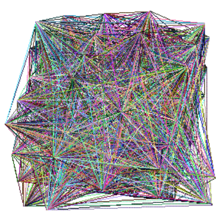
- Power law in the degree distribution [Faloutsos x 3 SIGCOMM99]

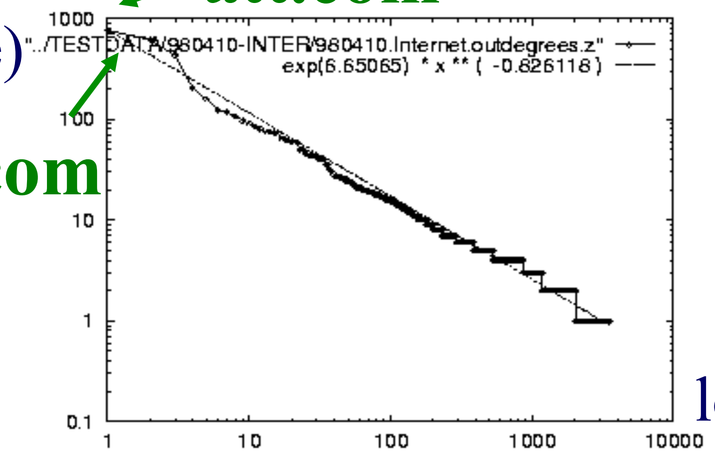
internet domains

att.com

log(degree)

ibm.com





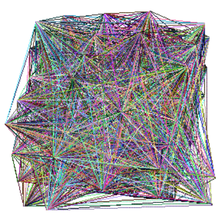
log(rank)

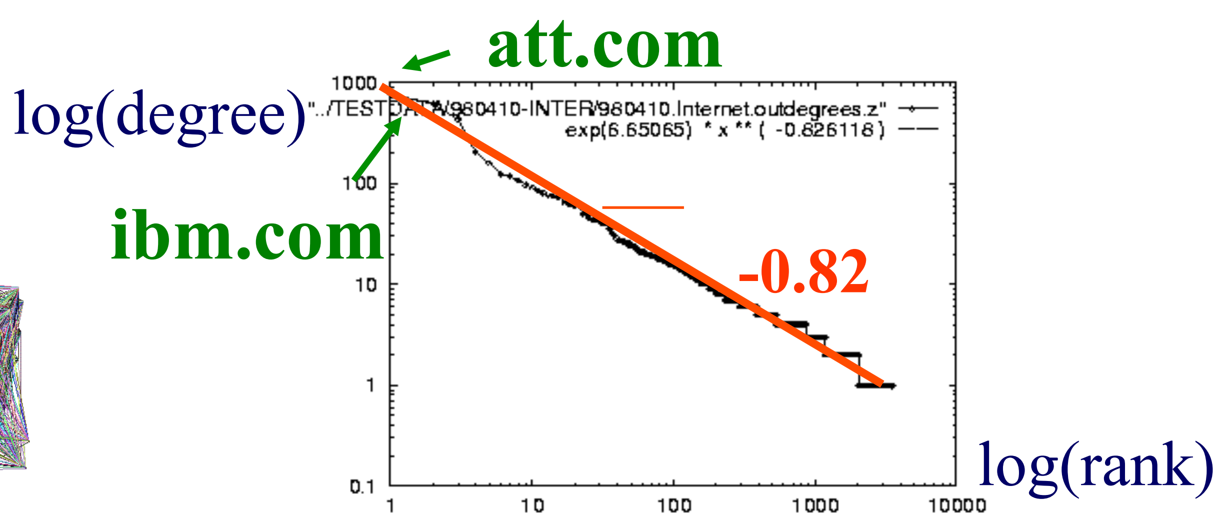
Carnegie Mellon

Solution# S.1

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

internet domains

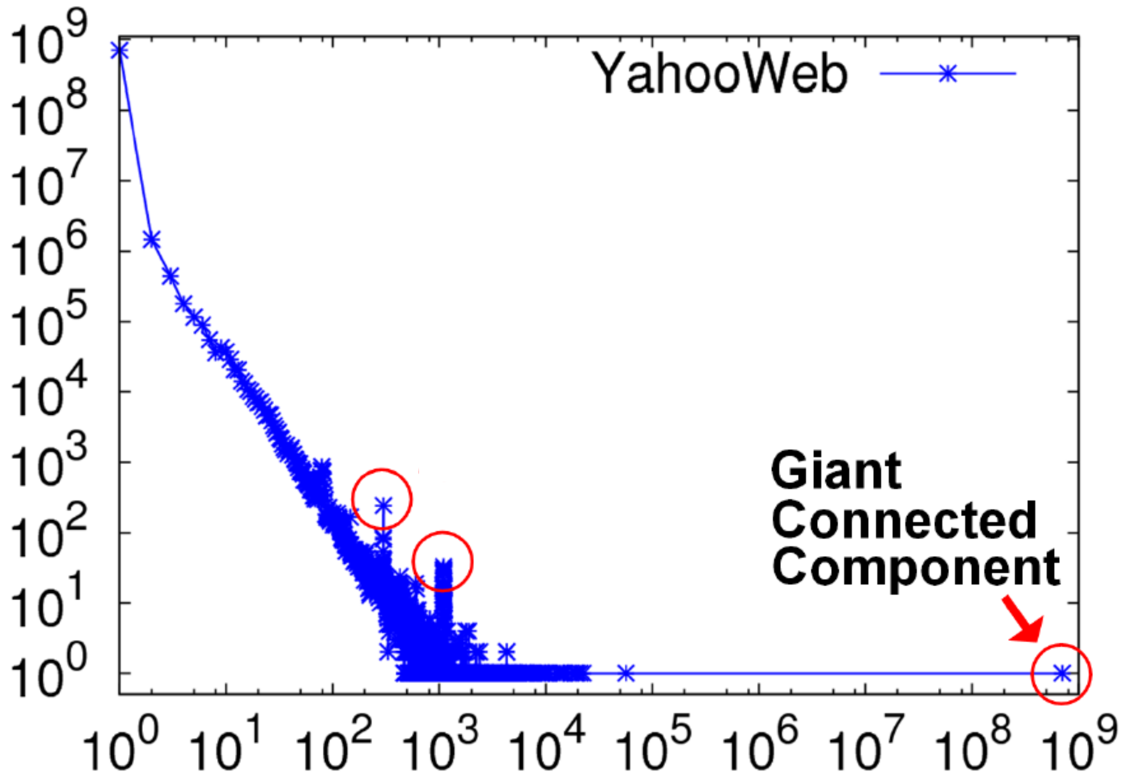




S2: connected component sizes

- Connected Components – 4 observations:

Count

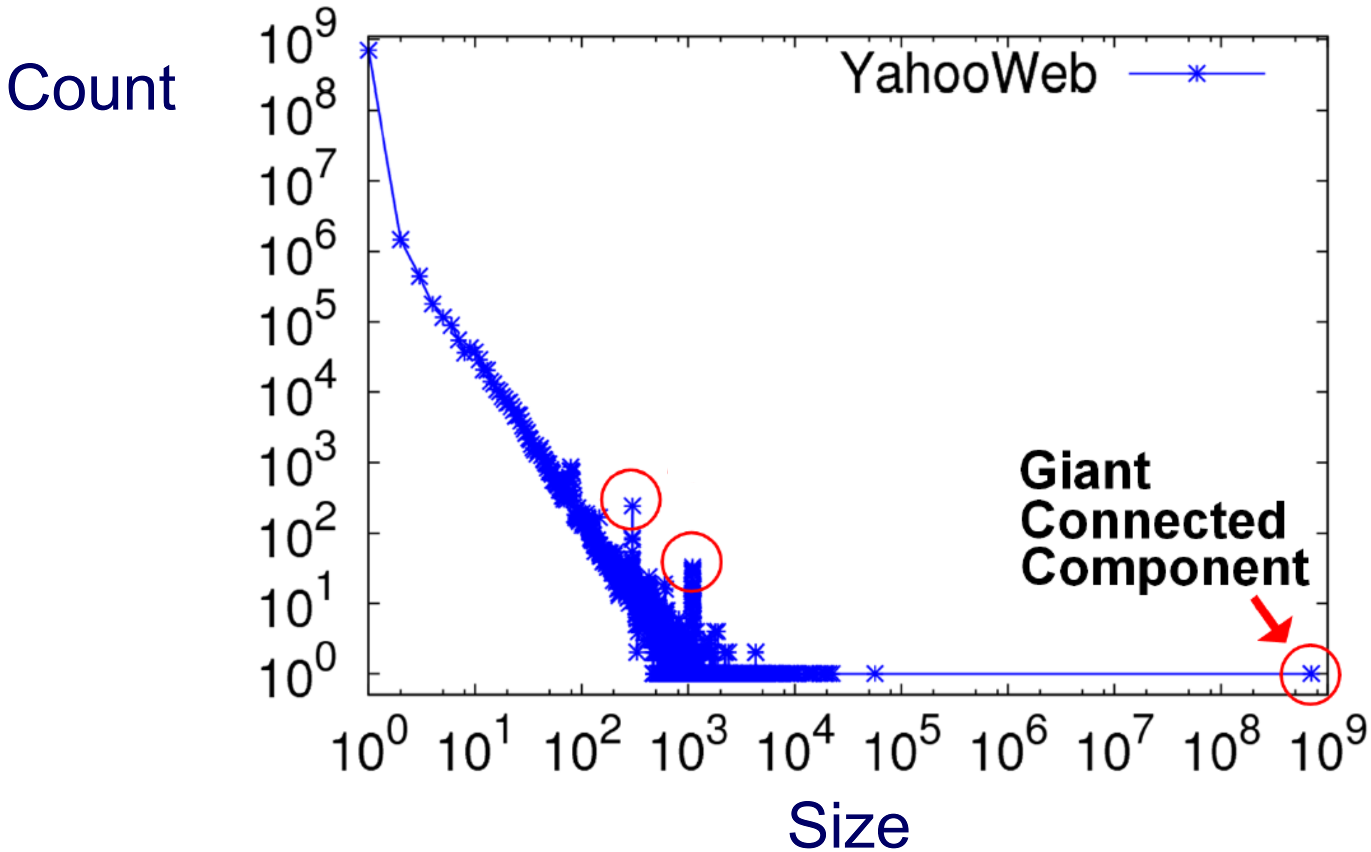


Size

1.4B nodes
6B edges

S2: connected component sizes

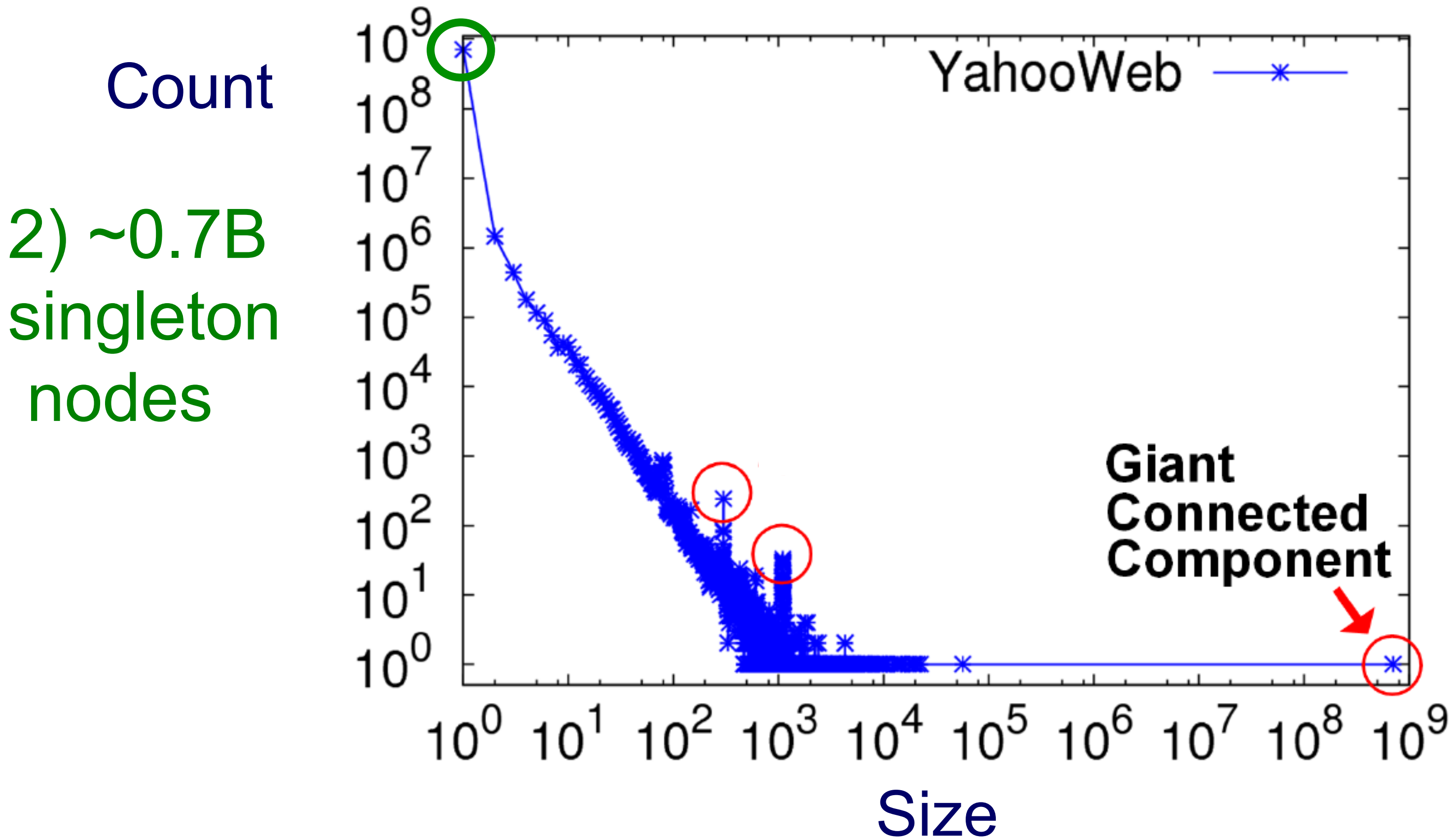
- Connected Components



1) 10K x larger than next

S2: connected component sizes

- Connected Components

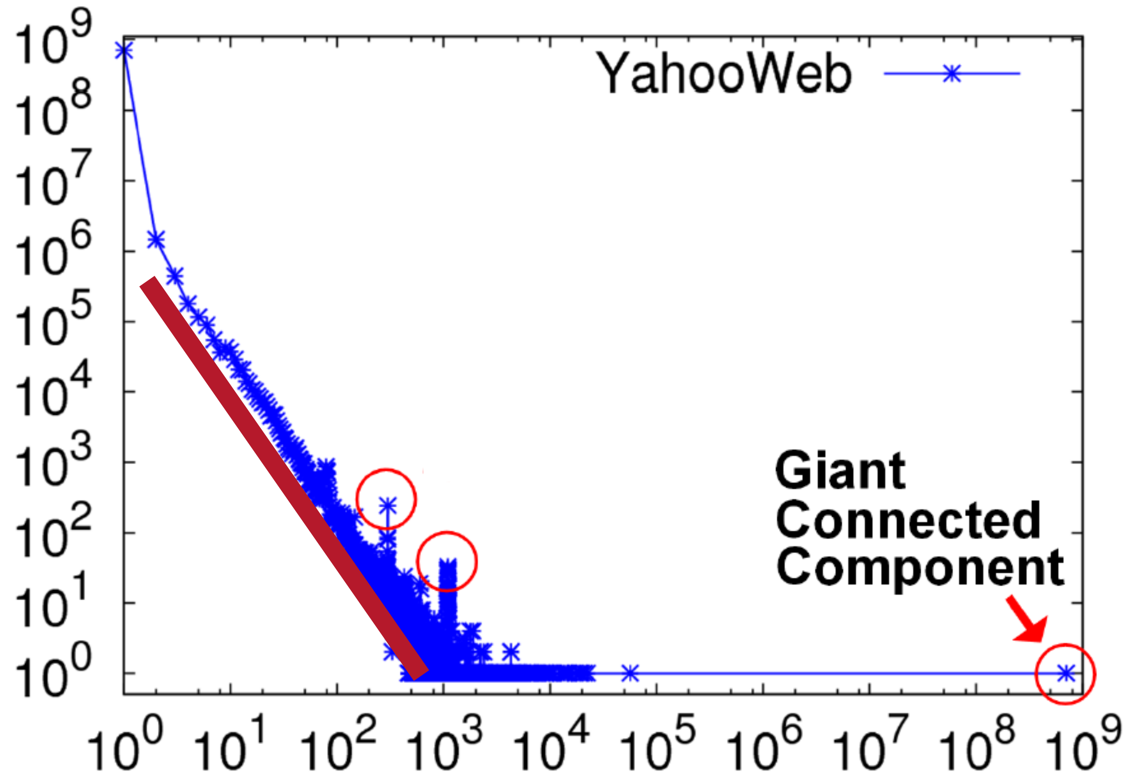


S2: connected component sizes

- Connected Components

Count

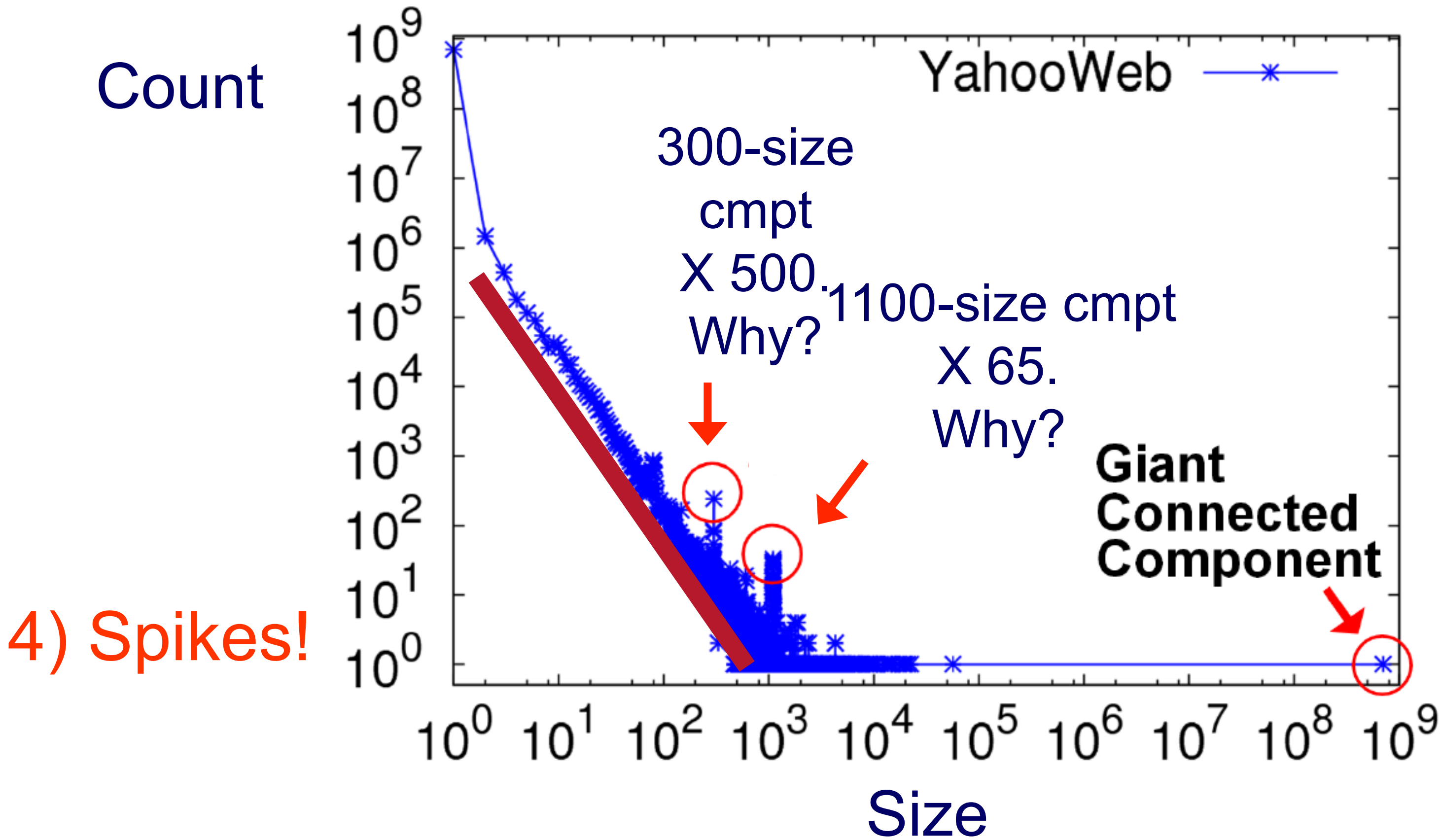
3) SLOPE!



Size

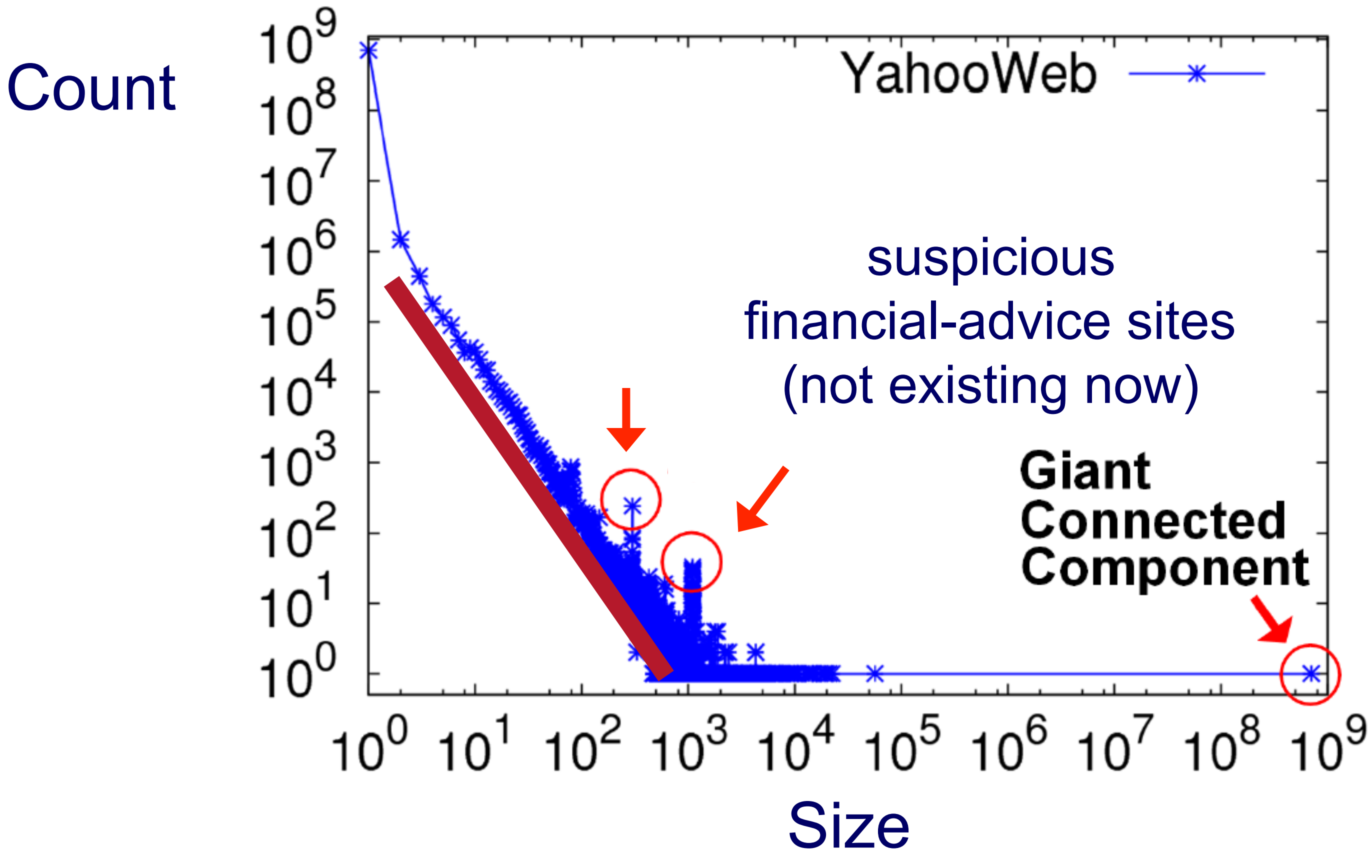
S2: connected component sizes

- Connected Components

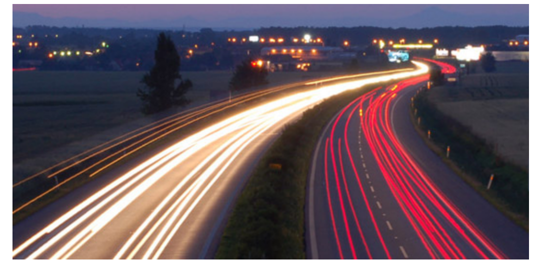


S2: connected component sizes

- Connected Components



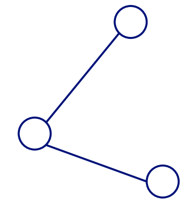
Roadmap



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

Carnegie Mellon

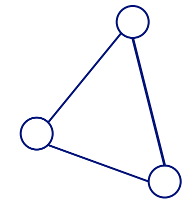
Solution# S.3: Triangle 'Laws'



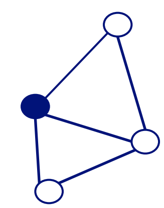
- Real social networks have a lot of triangles

Carnegie Mellon

Solution# S.3: Triangle 'Laws'

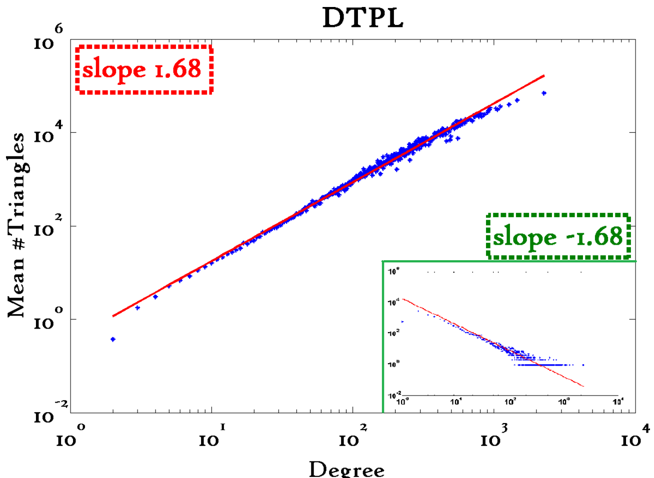


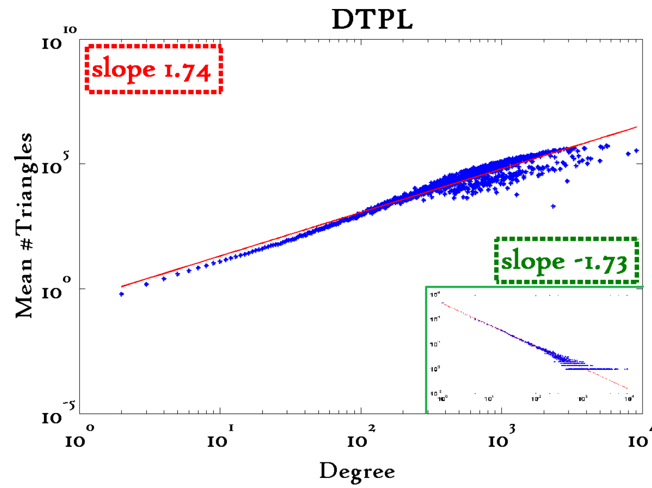
- Real social networks have a lot of triangles
  - Friends of friends are friends
- Any patterns?
  - 2x the friends, 2x the triangles ?



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

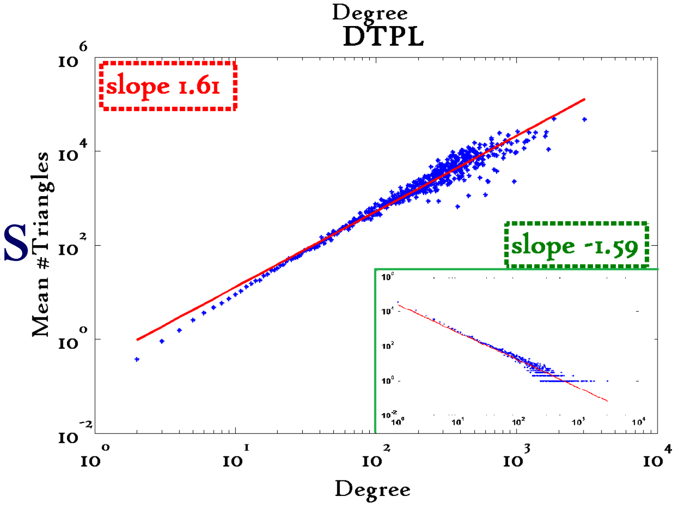
Reuters

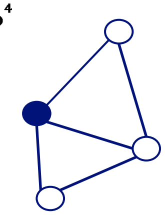




SN

Epinions



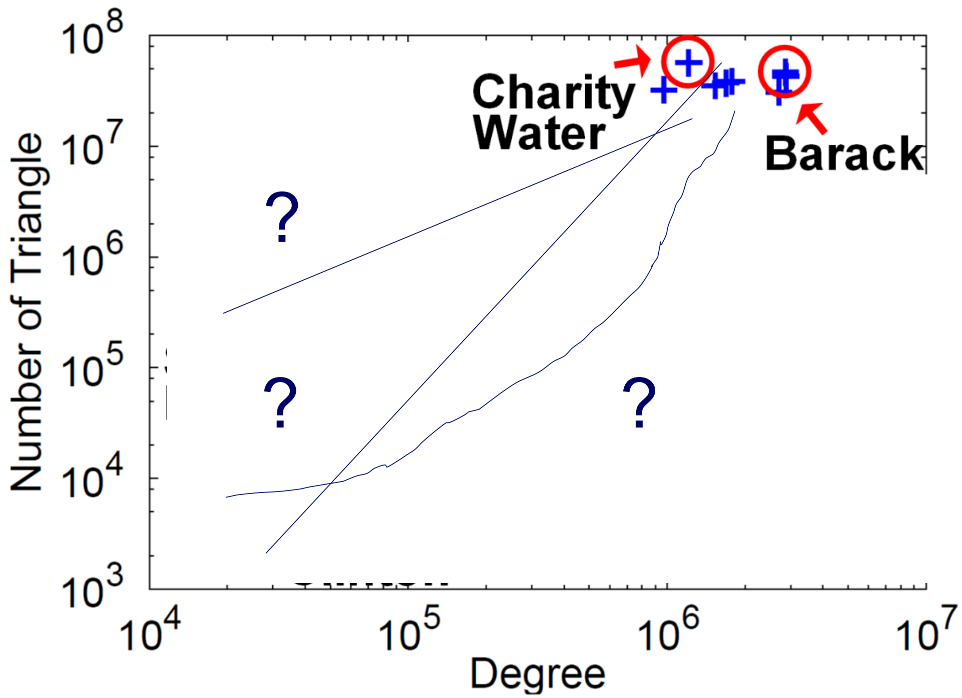


X-axis: degree

Y-axis: mean # triangles

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

Triangle counting for large graphs?

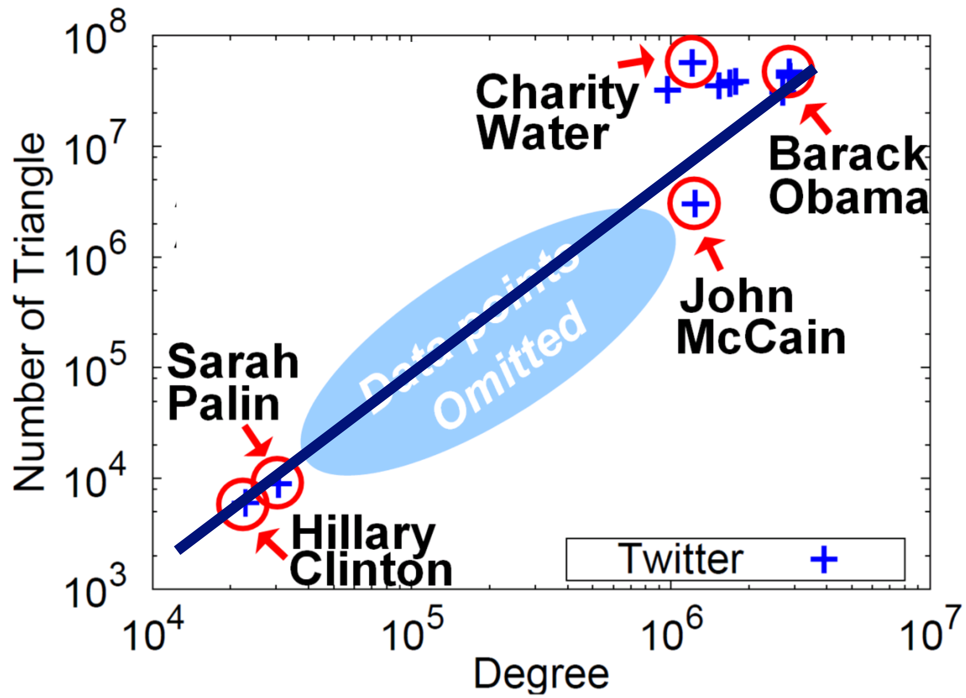


Anomalous nodes in Twitter(~ 3 billion edges)

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

Carnegie Mellon

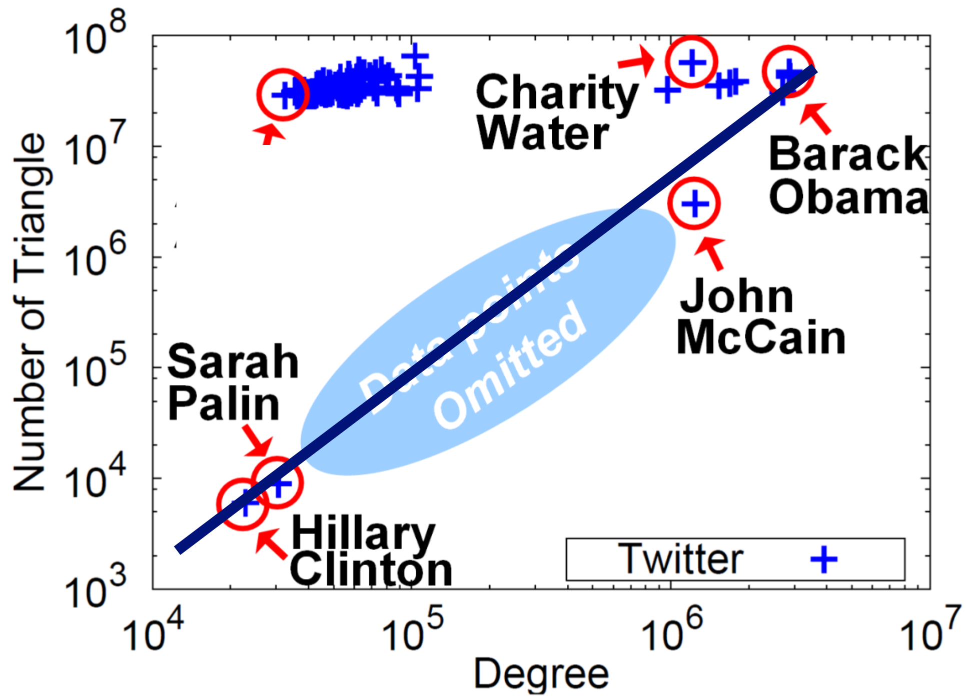
Triangle counting for large graphs?



Anomalous nodes in Twitter(~ 3 billion edges)

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

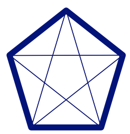
Triangle counting for large graphs?

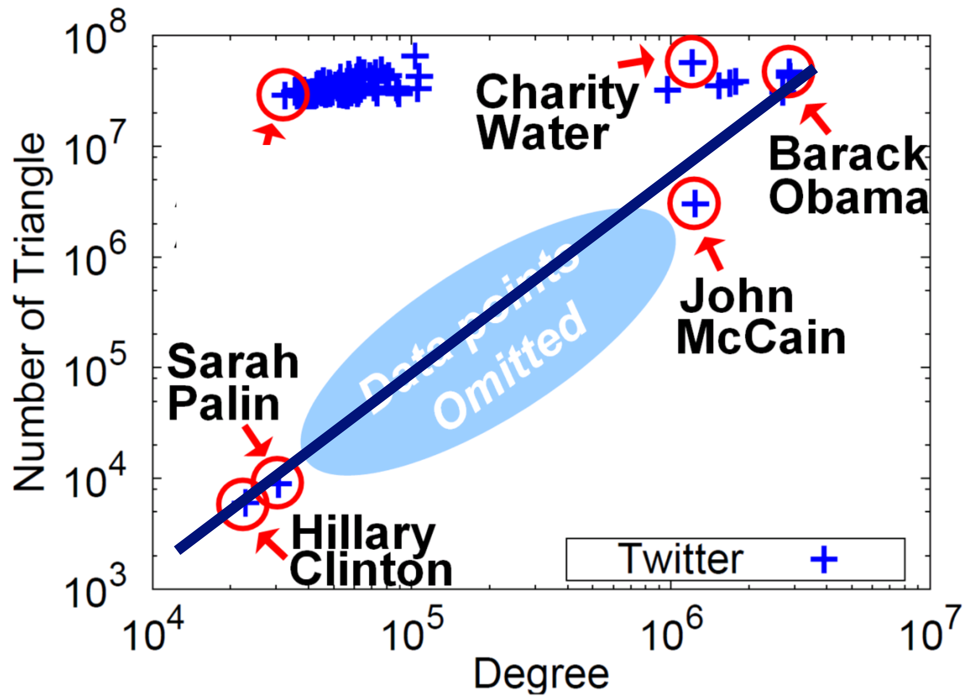


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?

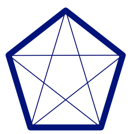


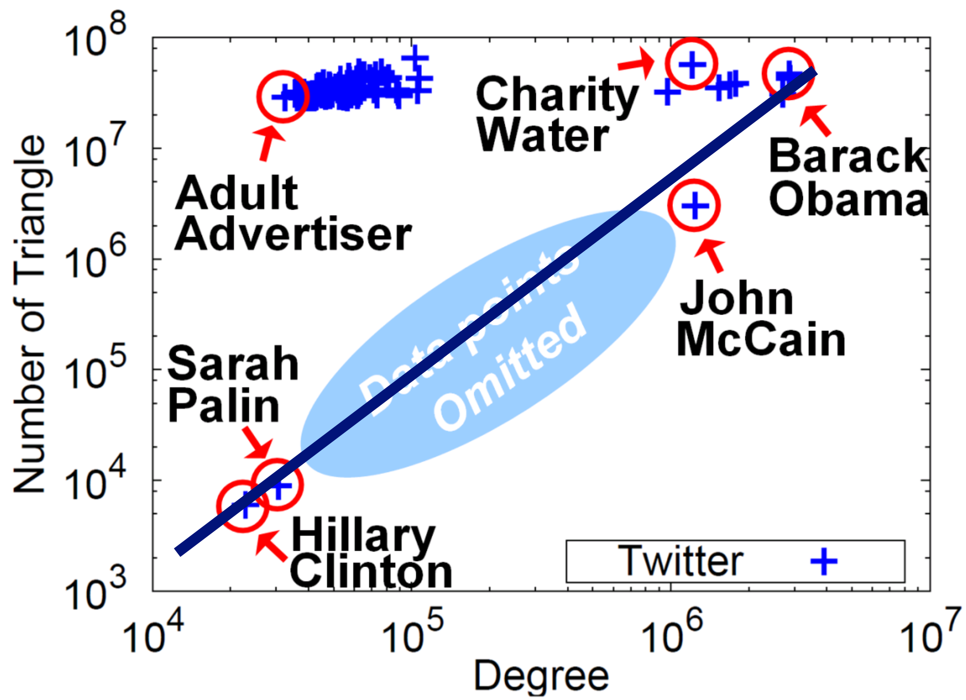


Anomalous nodes in Twitter(~ 3 billion edges)

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

Triangle counting for large graphs?



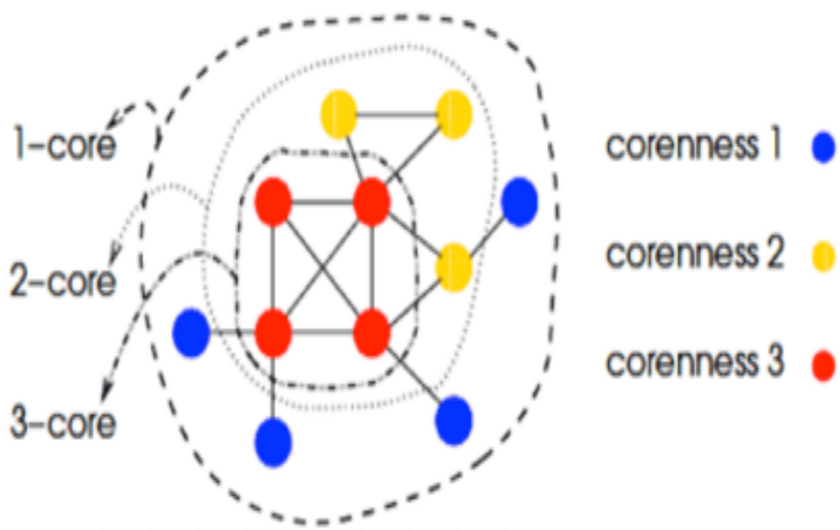


Anomalous nodes in Twitter(~ 3 billion edges)

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

S4: k-core patterns - dfn

- **k-core** (of a graph)
- **degeneracy** (of a graph)
- **coreness** (of a vertex)



Carnegie Mellon

CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies, and Algorithms

ICDM'16 (to appear)

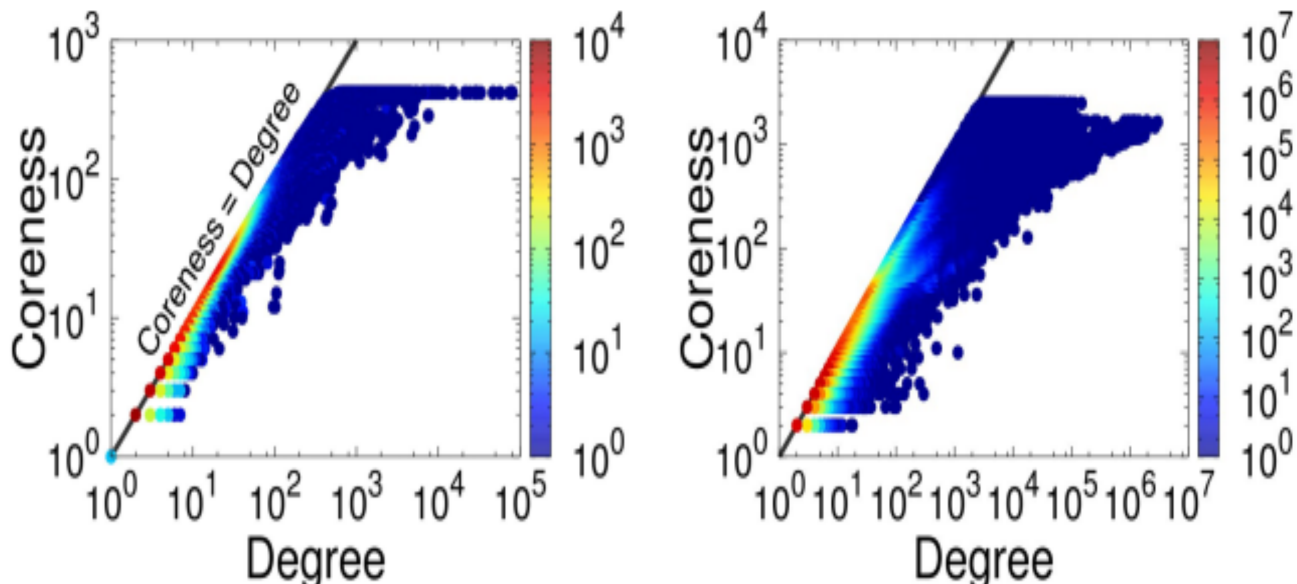
Kijung Shin, Tina Eliassi-Rad and CF

Carnegie Mellon

Mirror Pattern: Observation

– **coreness** (of a vertex): maximum k such that the vertex belongs to the k-core

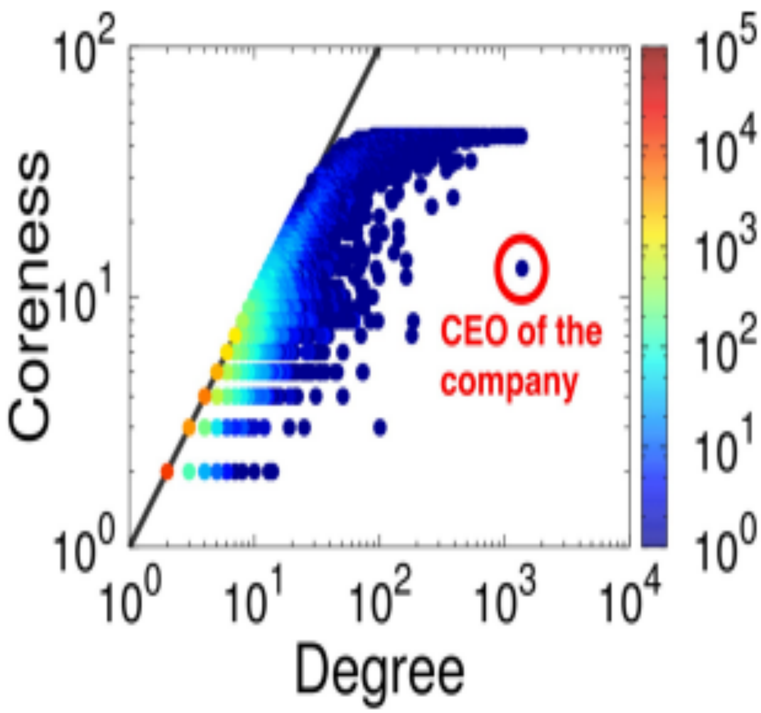
– Definition: **[Mirror Pattern]** *degree ~ coreness*



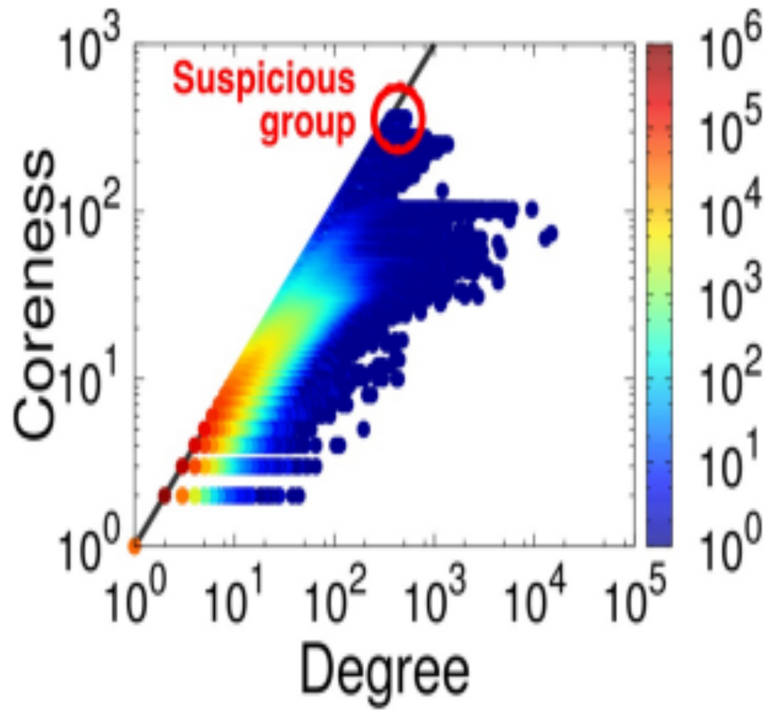
Rank correlation $\rho = 0.99$

Mirror Pattern: Application

- Exceptions are ‘strange’



Email ($\rho = 0.98$)



LiveJournal ($\rho = 0.99$)

MORE Graph Patterns

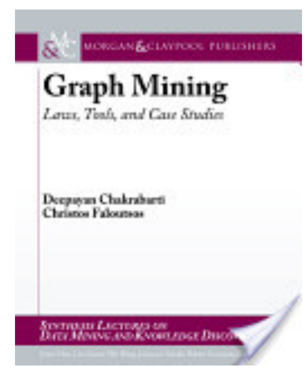
	Unweighted	Weighted
Static	L01. Power-law degree distribution [Faloutsos et al. `99, Kleinberg et al. `99, Chakrabarti et al. `04, Newman `04] L02. Triangle Power Law (TPL) [Tsourakakis `08] L03. Eigenvalue Power Law (EPL) [Siganos et al. `03] L04. Community structure [Flake et al. `02, Girvan and Newman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

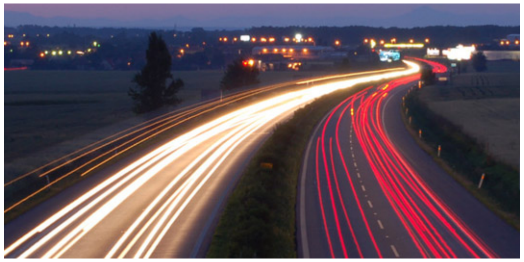
	Unweighted	Weighted
Static	**L01.** Power-law degree distribution [Faloutsos et al. \`99, Kleinberg et al. \`99, Chakrabarti et al. \`04, Newman \`04] **L02.** Triangle Power Law (TPL) [Tsourakakis \`08] **L03.** Eigenvalue Power Law (EPL) [Siganos et al. \`03] **L04.** Community structure [Flake et 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.

Roadmap



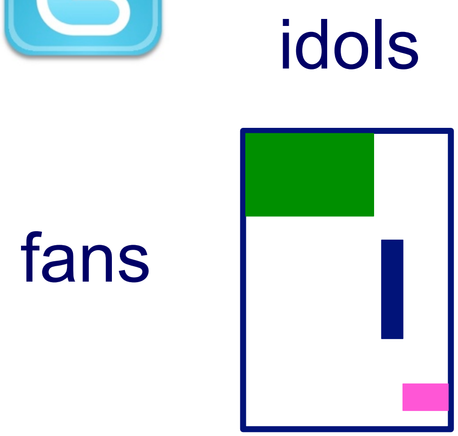
- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

Patterns

anomalies

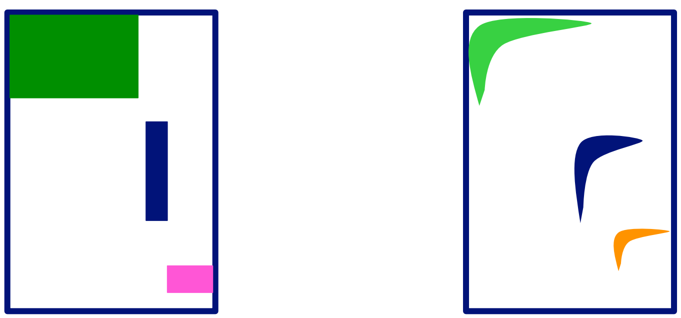
How to find 'suspicious' groups?

- 'blocks' are normal, right?



Except that:

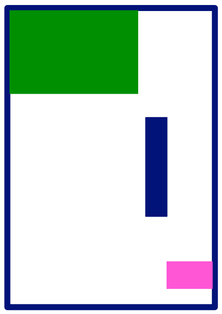
- ‘blocks’ are normal, right?
- ‘hyperbolic’ communities are more realistic [Araujo+, PKDD’14]

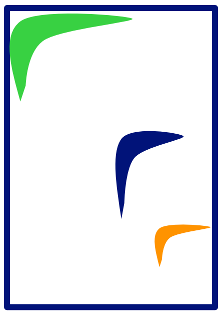


Except that:

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

Q: Can we spot blocks, easily?



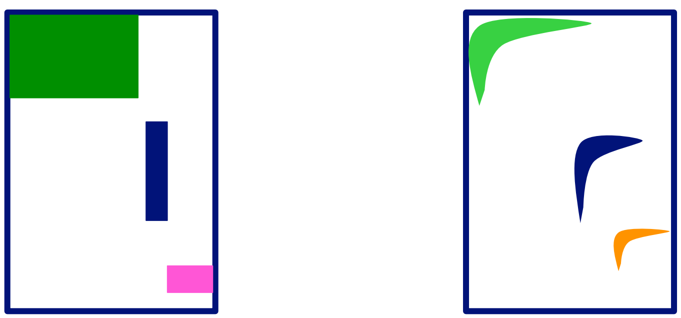


Except that:

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

Q: Can we spot blocks, easily?

A: Silver bullet: SVD!

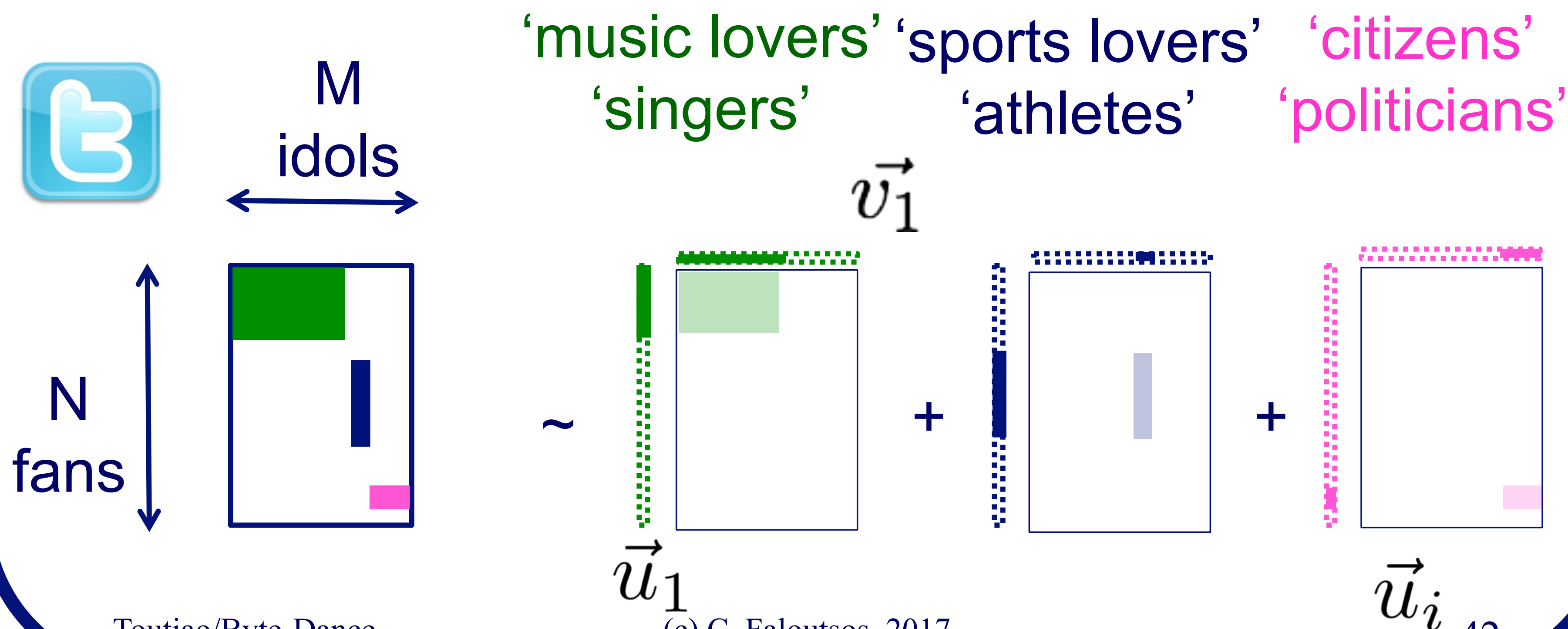


Carnegie Mellon

DETAILS

Crush intro to SVD

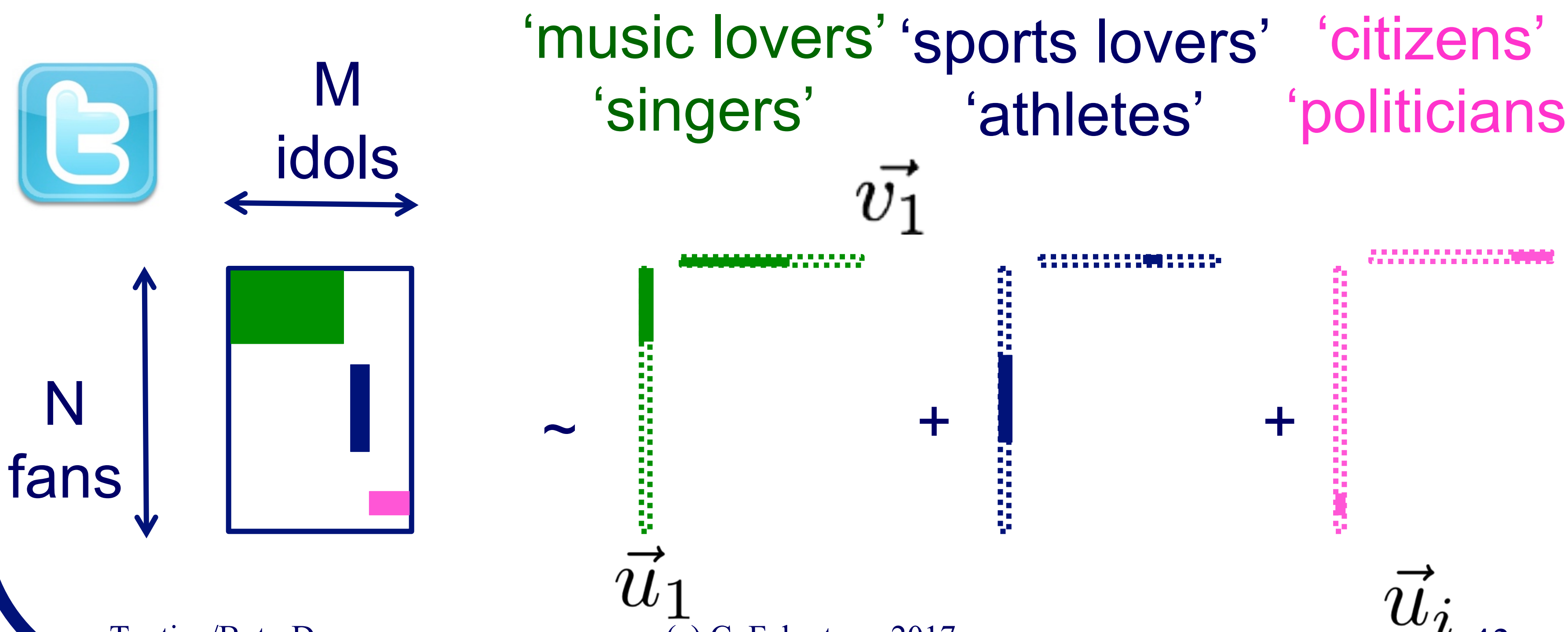
- Recall: (SVD) matrix factorization: finds blocks



DETAILS

Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks



Inferring Strange Behavior from Connectivity Pattern in Social Networks
PAKDD'14

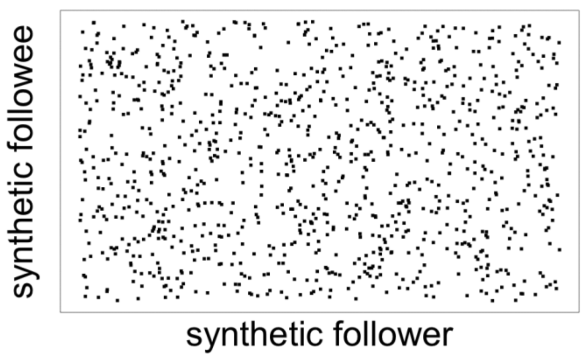


Meng Jiang, Peng Cui, Shiqiang Yang (Tsinghua)
Alex Beutel, Christos Faloutsos (CMU)

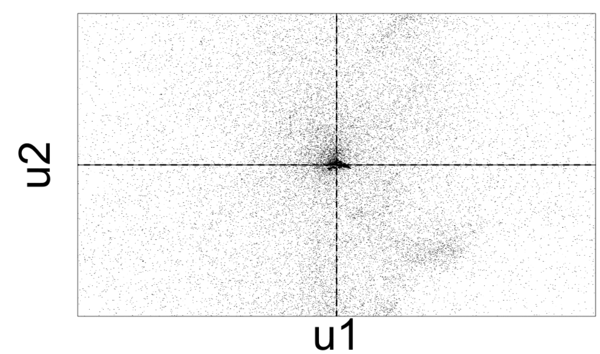
Lockstep and *Spectral Subspace Plot*

- Case #0: No lockstep behavior in random power law graph of 1M nodes, 3M edges
- Random ⟷ “Scatter”

Adjacency Matrix



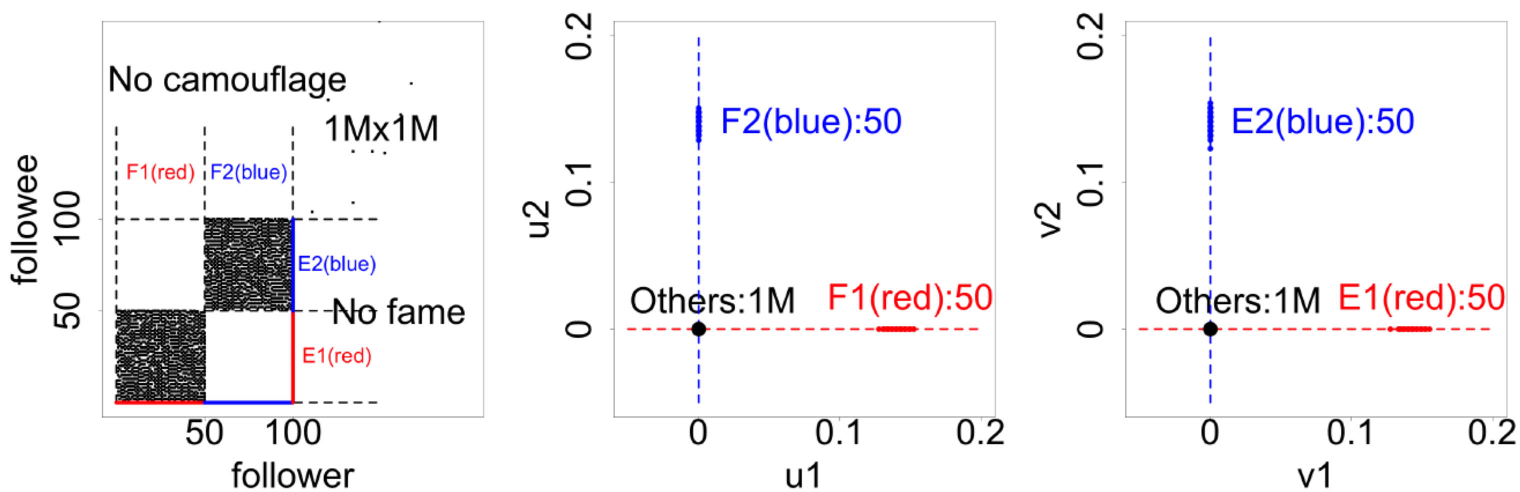
Spectral Subspace Plot



Lockstep and *Spectral Subspace Plot*

- Case #1: non-overlapping lockstep
- "Blocks" ⟷ "Rays"

Adjacency Matrix | Spectral Subspace Plot

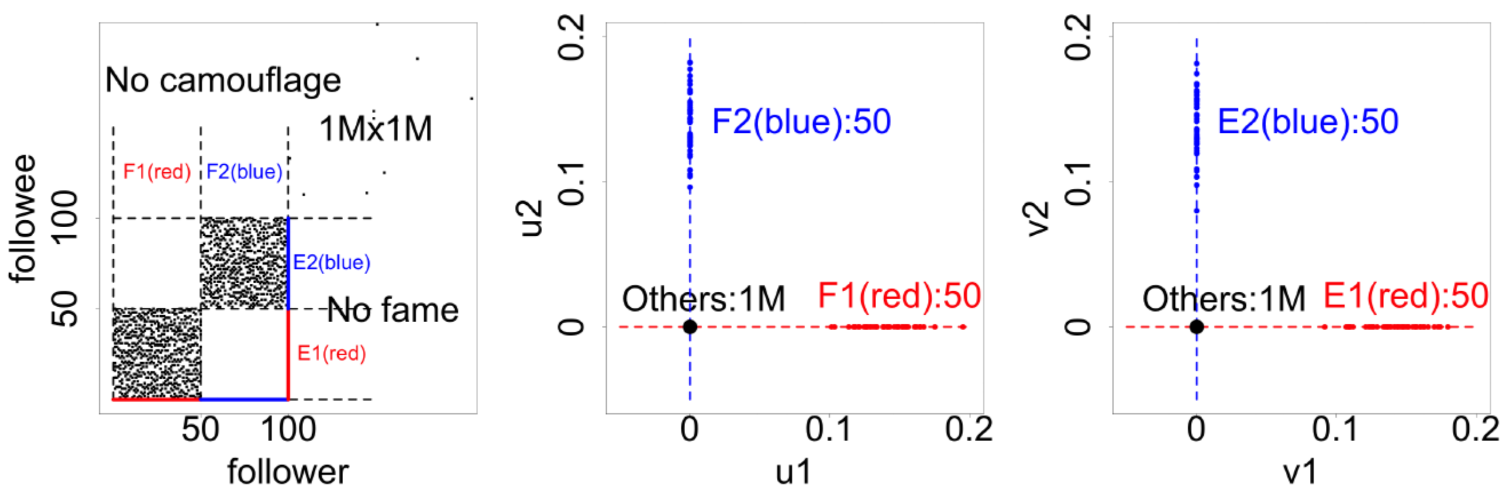


Rule 1 (short "rays"): two blocks, high density (90%), no "camouflage", no "fame"

Lockstep and *Spectral Subspace Plot*

- Case #2: non-overlapping lockstep
- "Blocks; low density" ⟷ Elongation

Adjacency Matrix — Spectral Subspace Plot

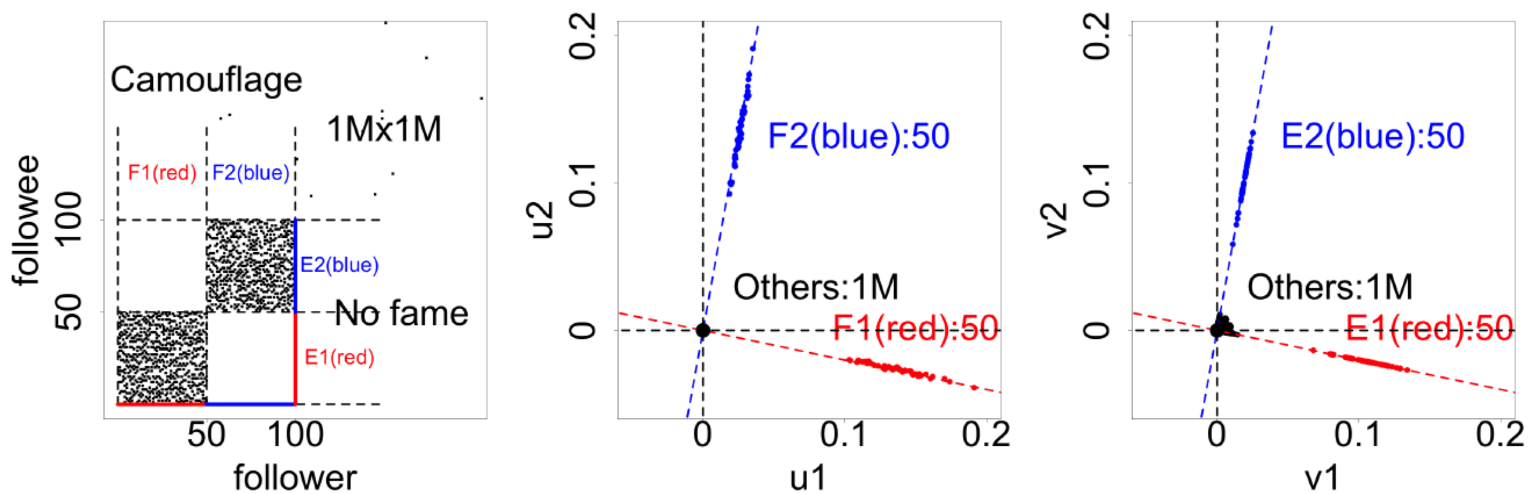


Rule 2 (long "rays"): two blocks, low density (50%), no "camouflage", no "fame"

Lockstep and *Spectral Subspace Plot*

- Case #3: non-overlapping lockstep
- "**Camouflage**" (or "Fame") ⟷ Tilting "Rays"

Adjacency Matrix　　　Spectral Subspace Plot



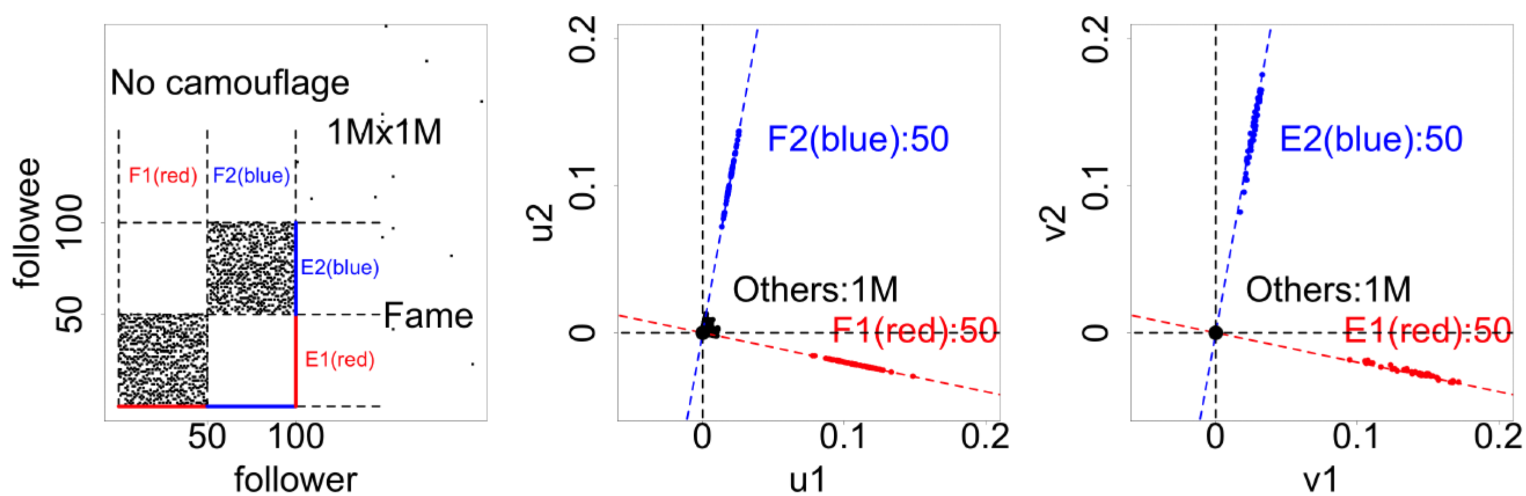
Rule 3 (tilting "rays"): two blocks, with "camouflage", no "fame"

Carnegie Mellon

Lockstep and *Spectral Subspace Plot*

- Case #3: non-overlapping lockstep
- "Camouflage" (or "**Fame**") ⟷ Tilting "Rays"

Adjacency Matrix Spectral Subspace Plot



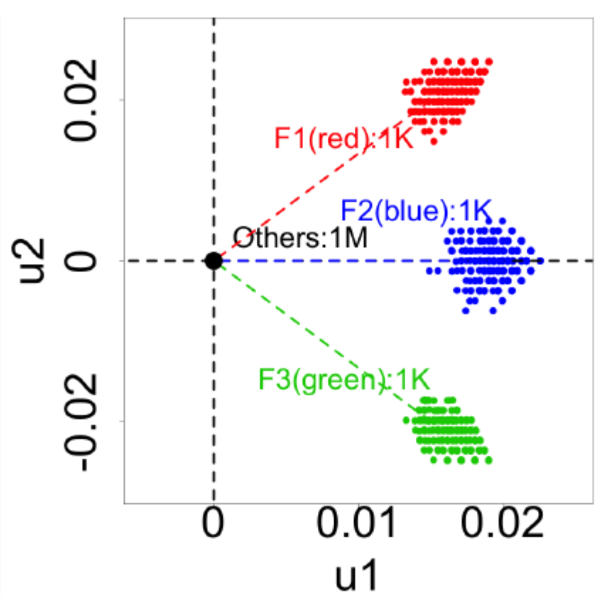
Rule 3 (tilting "rays"): two blocks, no "camouflage", with "fame"

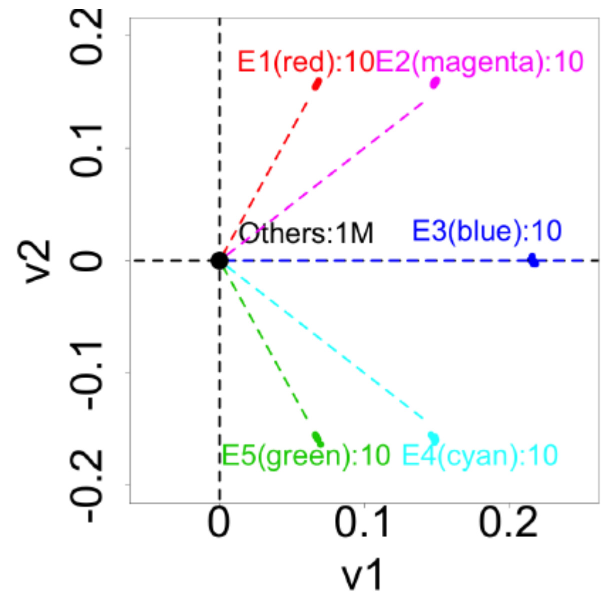
Lockstep and *Spectral Subspace Plot*

- Case #4: ? lockstep
- "?" ⟷ "Pearls"

Adjacency Matrix Spectral Subspace Plot

?

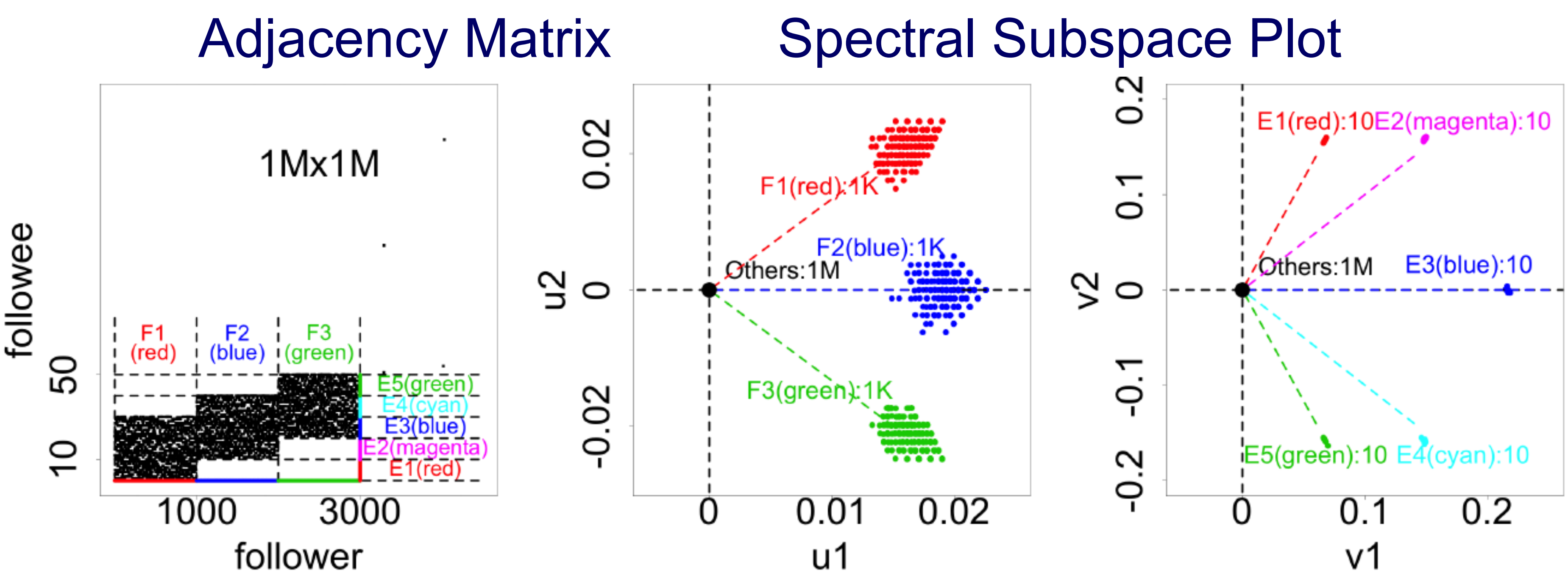




Carnegie Mellon

Lockstep and *Spectral Subspace Plot*

- Case #4: **overlapping** lockstep
- “**Staircase**” ⟷ “Pearls”



Rule 4 (“pearls”): a “staircase” of three partially overlapping blocks.

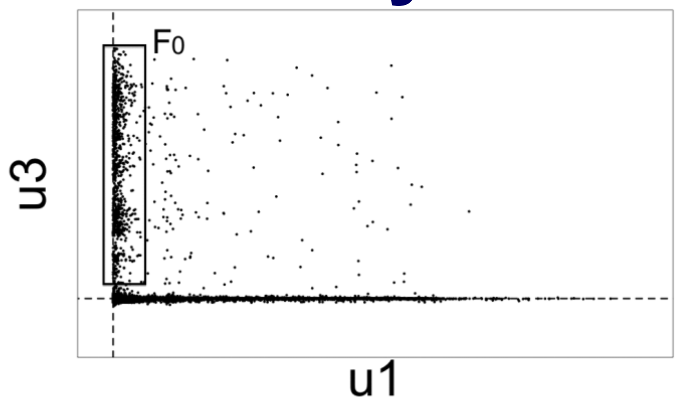
Carnegie Mellon

Dataset

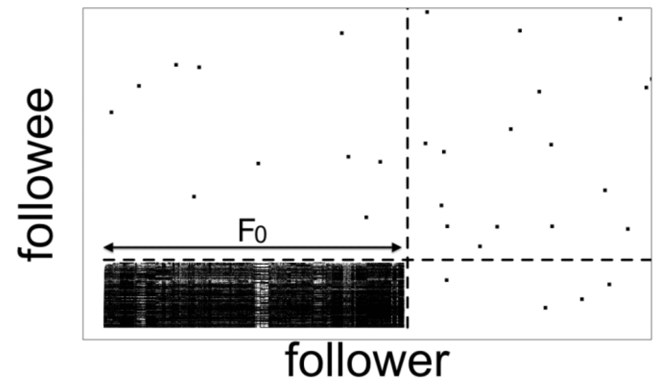
- Tencent Weibo
- 117 million nodes (with profile and UGC data)
- 3.33 billion directed edges

Real Data

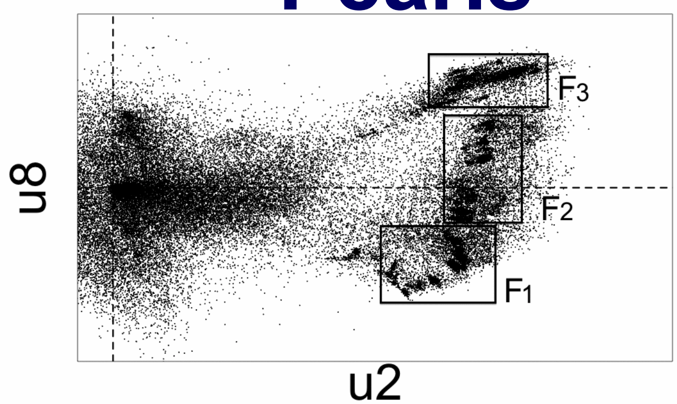
"Rays"



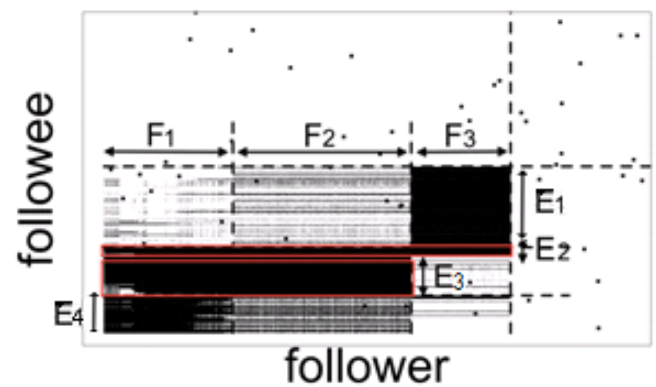
"Block"



"Pearls"

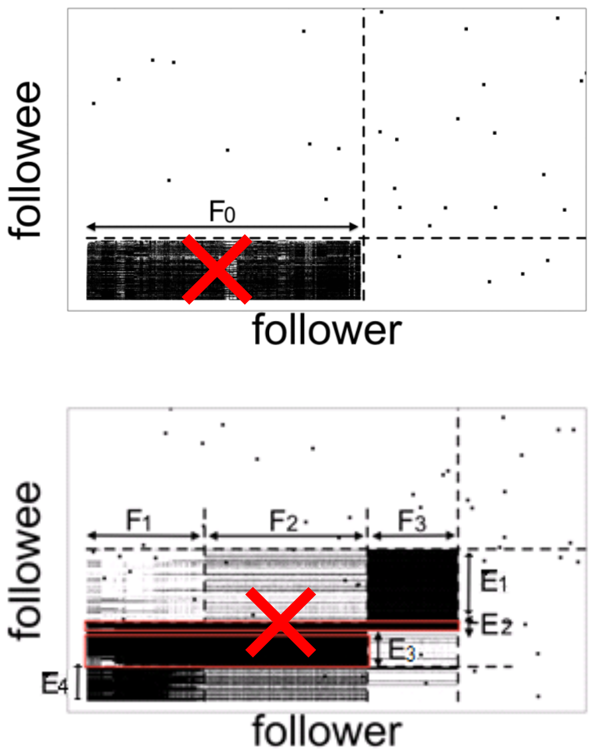


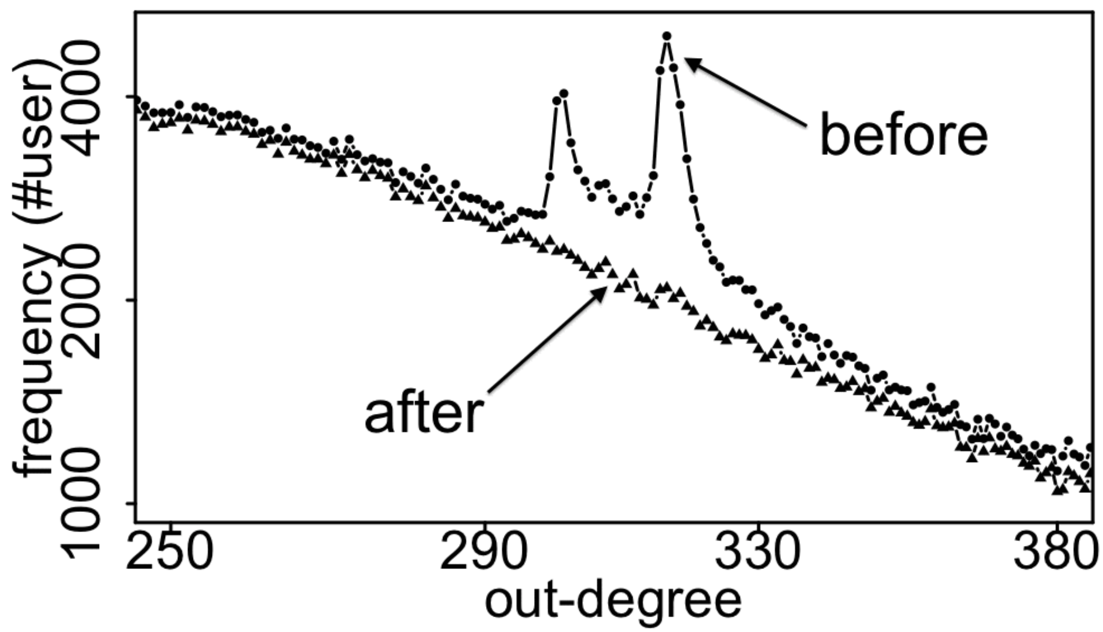
"Staircase"



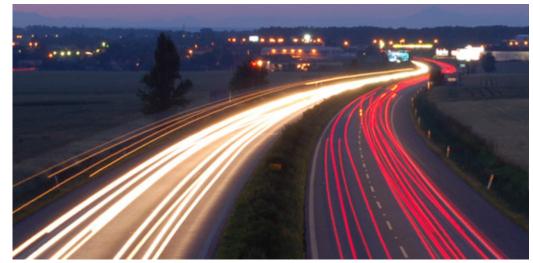
Real Data

- Spikes on the out-degree distribution





Roadmap

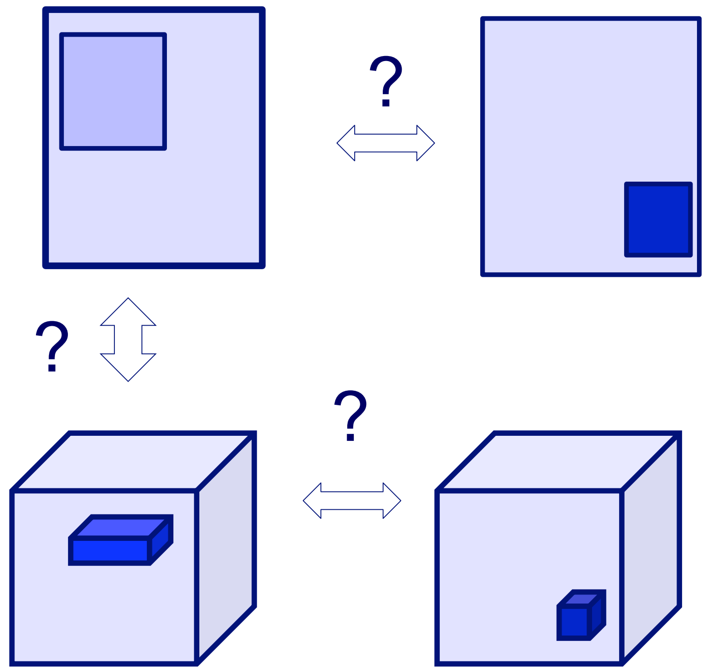


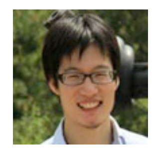
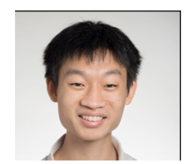
- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral methods
      - **Suspiciousness**
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

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.

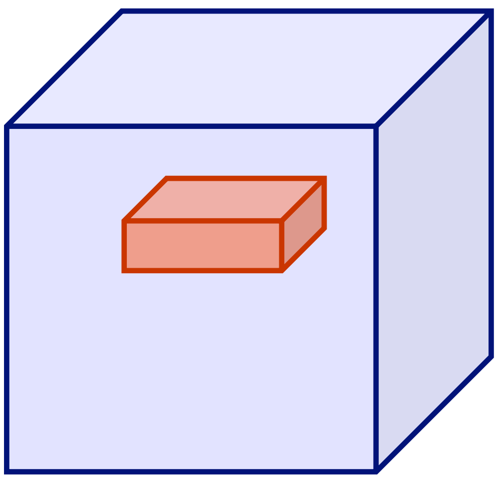
Suspicious Patterns in Event Data

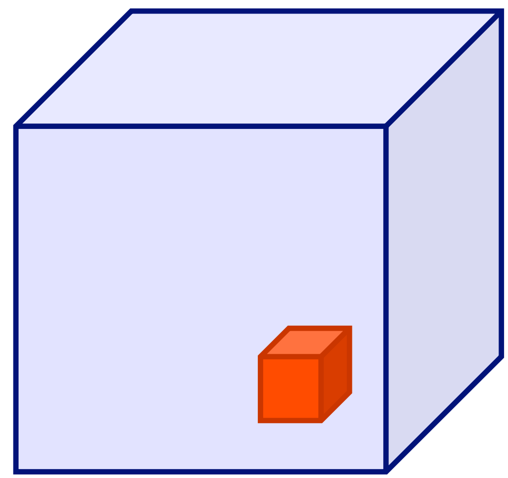
Which is more suspicious?

20,000 Users
Retweeting same 20 tweets
6 times each
All in 10 hours

vs.

225 Users
Retweeting same 1 tweet
15 times each
All in 3 hours
All from 2 IP addresses





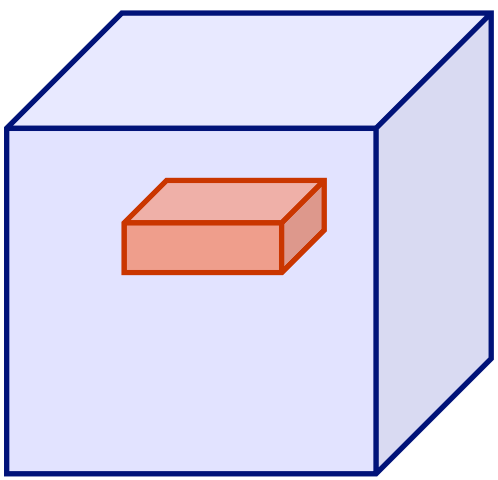
Suspicious Patterns in Event Data

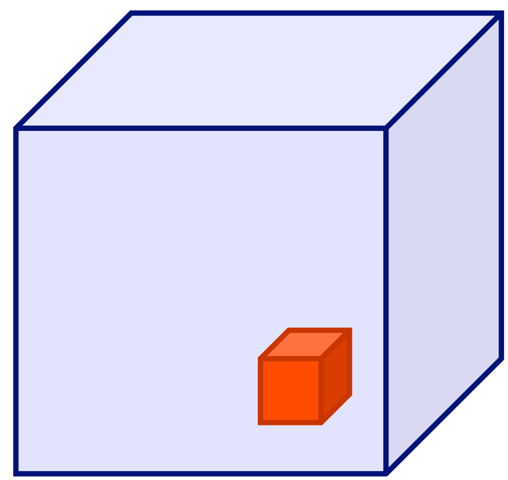
Which is more suspicious?

20,000 Users
Retweeting same 20 tweets
6 times each
All in 10 hours

vs.

225 Users
Retweeting same 1 tweet
15 times each
All in 3 hours
All from 2 IP addresses





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

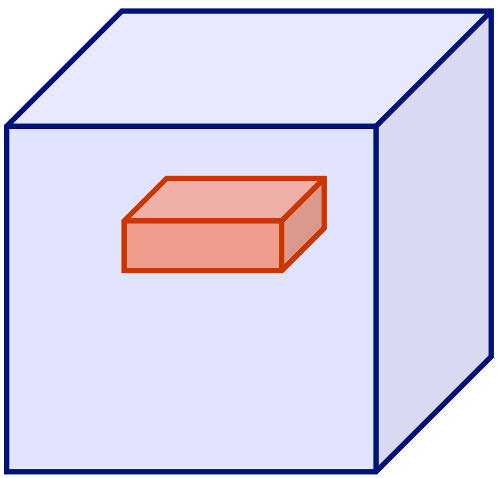
Carnegie Mellon

ICDM 2015

Suspicious Patterns in Event Data

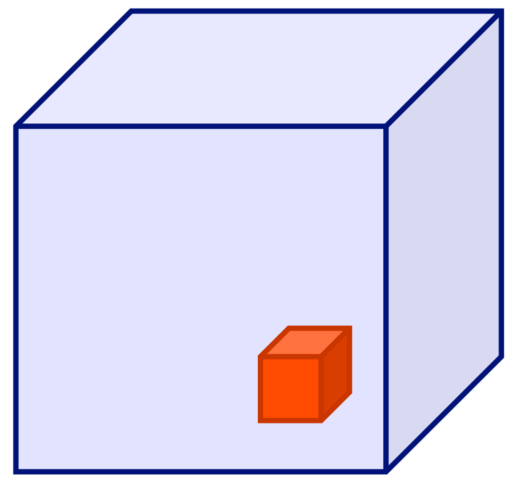
Which is more suspicious?

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



size

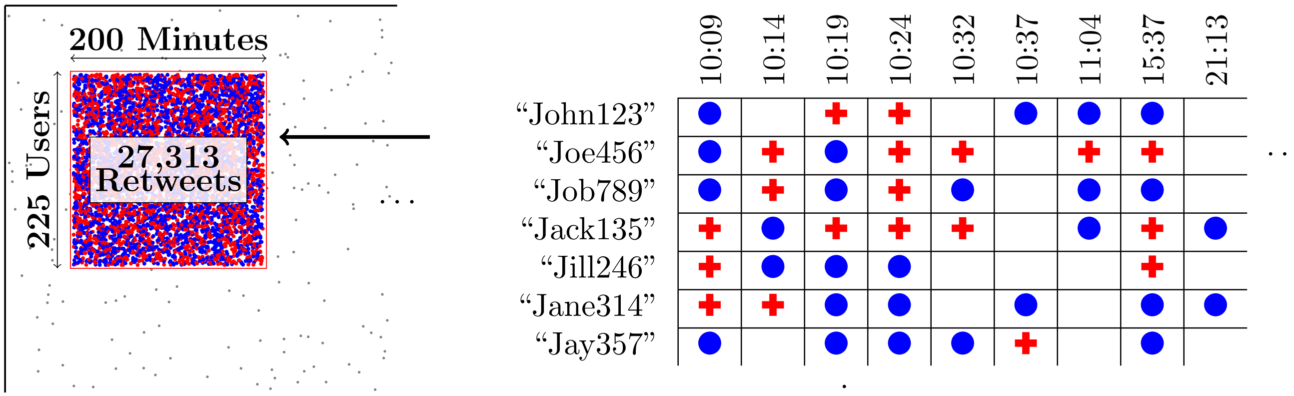
contrast



Toutiao

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

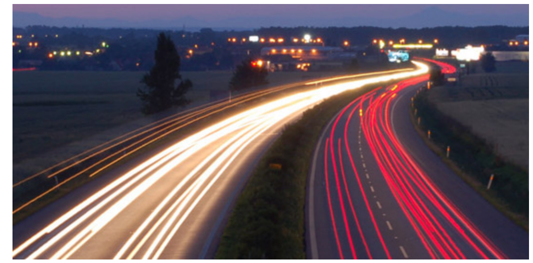
Suspicious Patterns in Event Data



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

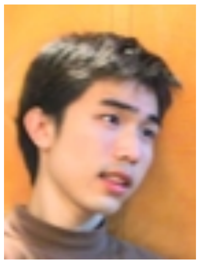
	#	User×tweet×IP×minute	Mass c	Suspiciousness
CROSSSPOT	1	14×1×2×1,114	41,396	1,239,865
	2	225×1×2×200	27,313	777,781
	3	8×2×4×1,872	17,701	491,323
HOSVD	1	24×6×11×439	3,582	131,113
	2	18×4×5×223	1,942	74,087
	3	14×2×1×265	9,061	381,211

Roadmap

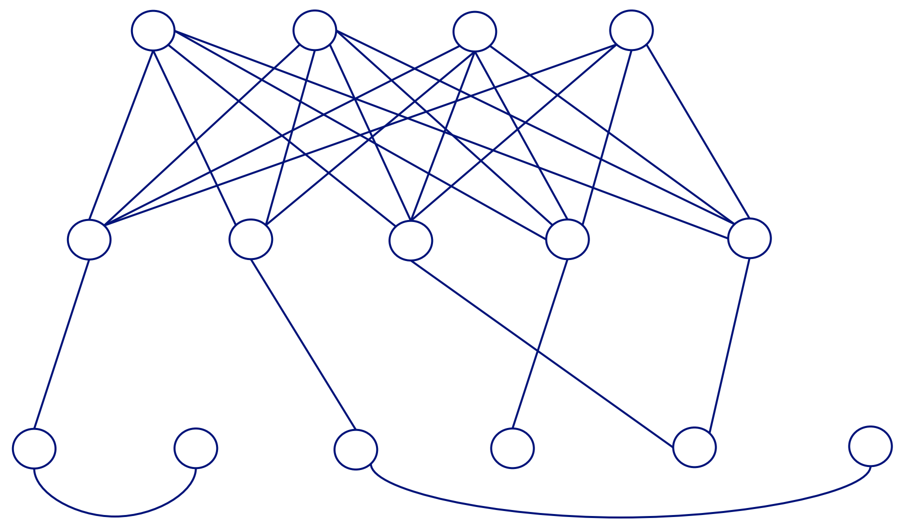


- Introduction – Motivation
- Part#1: Patterns in graphs
  - P1.1: Patterns
  - P1.2: Anomaly / fraud detection
    - No labels – spectral methods
    - With labels: Belief Propagation
- Part#2: time-evolving graphs; tensors
- Conclusions

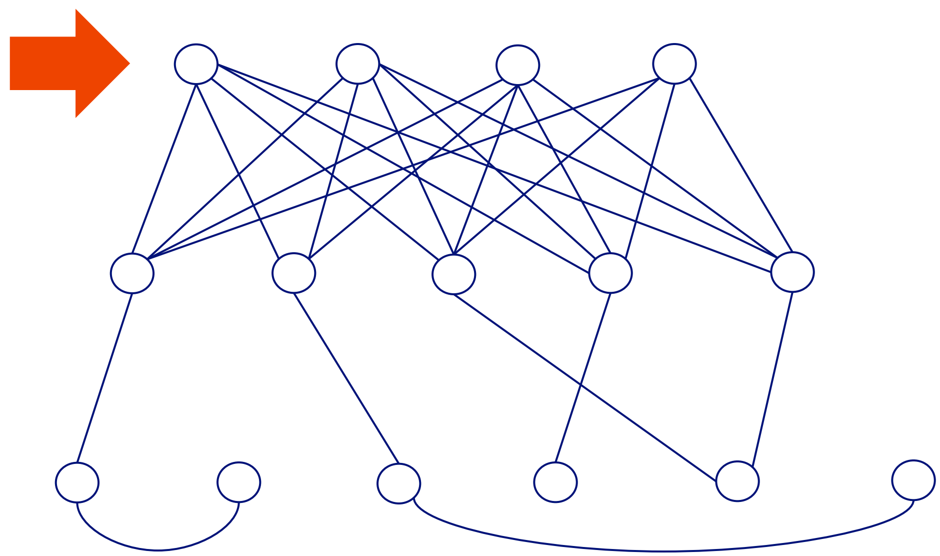
E-bay Fraud detection



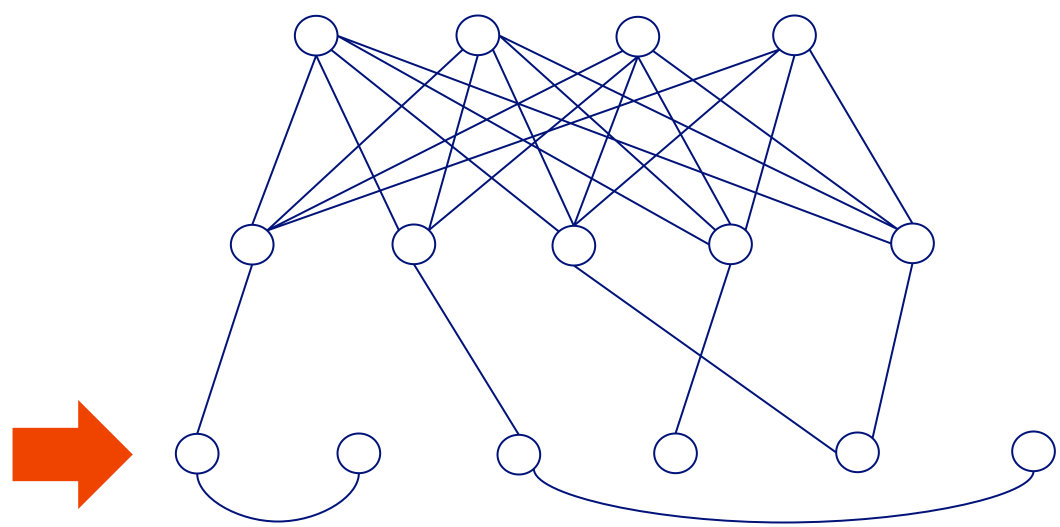
w/ Polo Chau &
Shashank Pandit, CMU
[www'07]



E-bay Fraud detection

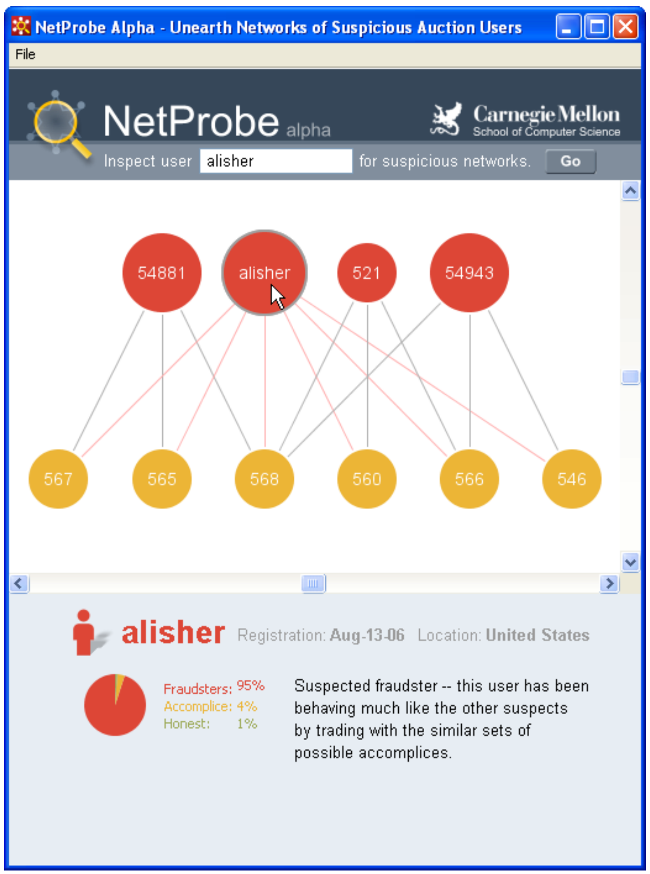


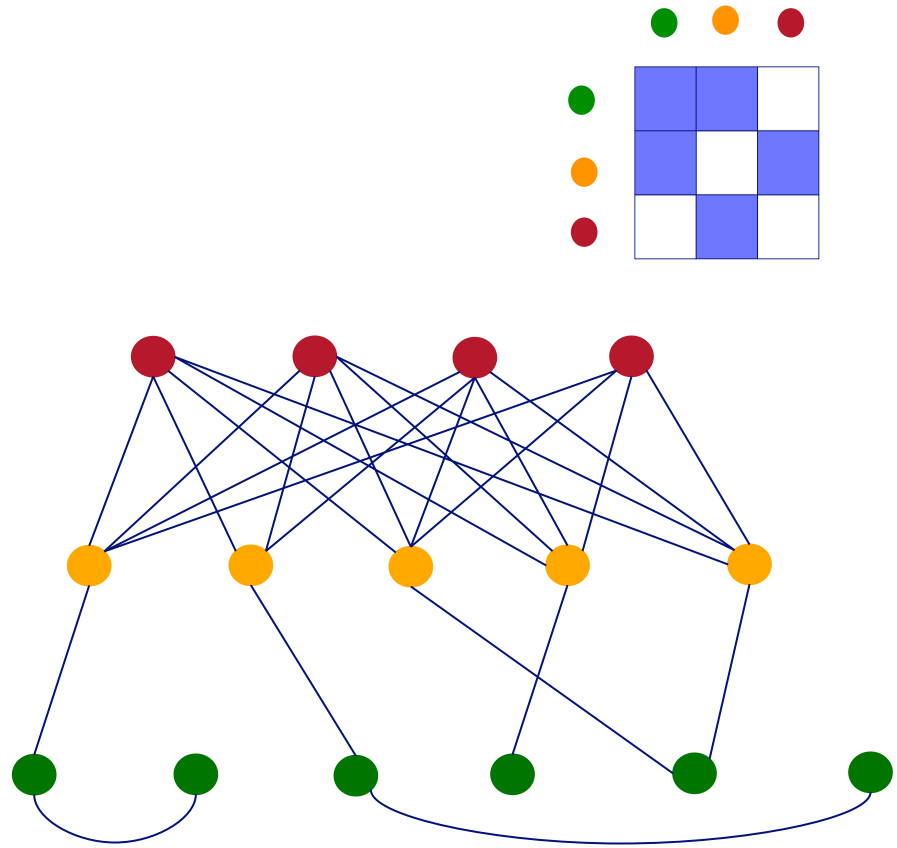
E-bay Fraud detection



Carnegie Mellon

E-bay Fraud detection - NetProbe





Popular press

USA TODAY

The Washington Post

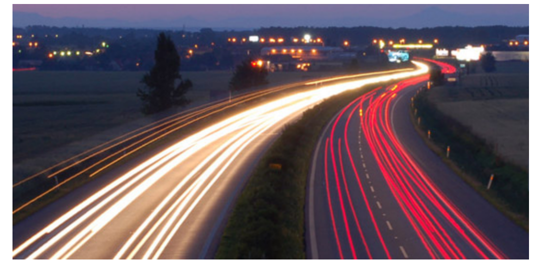
Los Angeles Times

And less desirable attention:

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

Carnegie Mellon

Roadmap



- Introduction – Motivation
- Part#1: Patterns in graphs
  - Patterns
  - Anomaly / fraud detection
    - No labels - Spectral methods
    - w/ labels: Belief Propagation – closed formulas
- Part#2: time-evolving graphs; tensors
- Conclusions

Carnegie Mellon

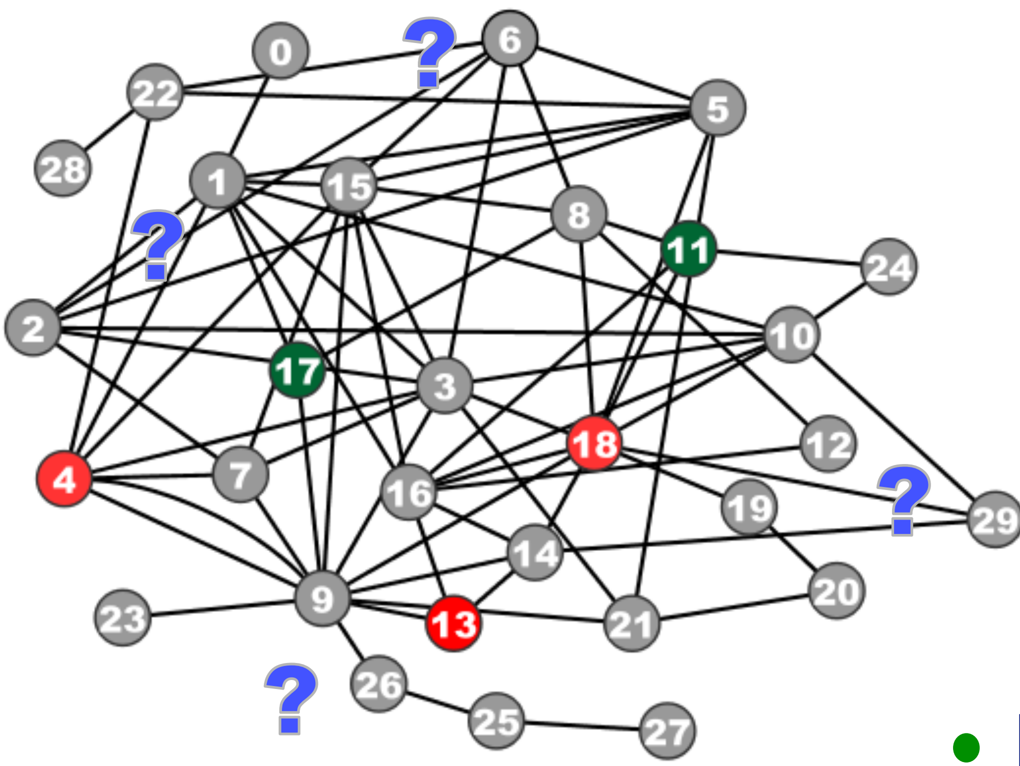
Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms

Danai Koutra
U Kang
Hsing-Kuo Kenneth Pao

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

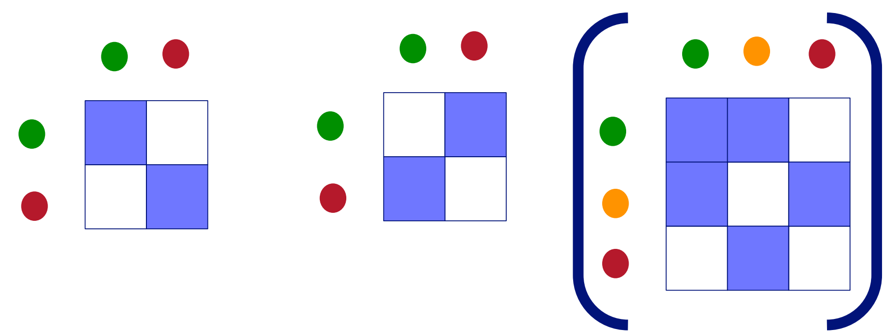
ECML PKDD, 5-9 September 2011, Athens, Greece

Problem Definition: GBA techniques



Given: Graph; &
few labeled nodes

Find: labels of rest
(assuming network effects)



Are they related?

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

Are they related? YES!

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

Correspondence of Methods

Method	Matrix		Unknown		known
RWR	$[\mathbf{I} - c\ \underline{\mathbf{A}}\mathbf{D}^{-1}]$	$\times$	$\mathbf{x}$	$=$	$(1-c)\mathbf{y}$
SSL	$[\mathbf{I} + a(\mathbf{D} - \underline{\mathbf{A}})]$	$\times$	$\mathbf{x}$	$=$	$\mathbf{y}$
FABP	$[\mathbf{I} + a\,\mathbf{D} - c'\underline{\mathbf{A}}]$	$\times$	$\mathbf{b_h}$	$=$	$\boldsymbol{\phi}_\mathbf{h}$

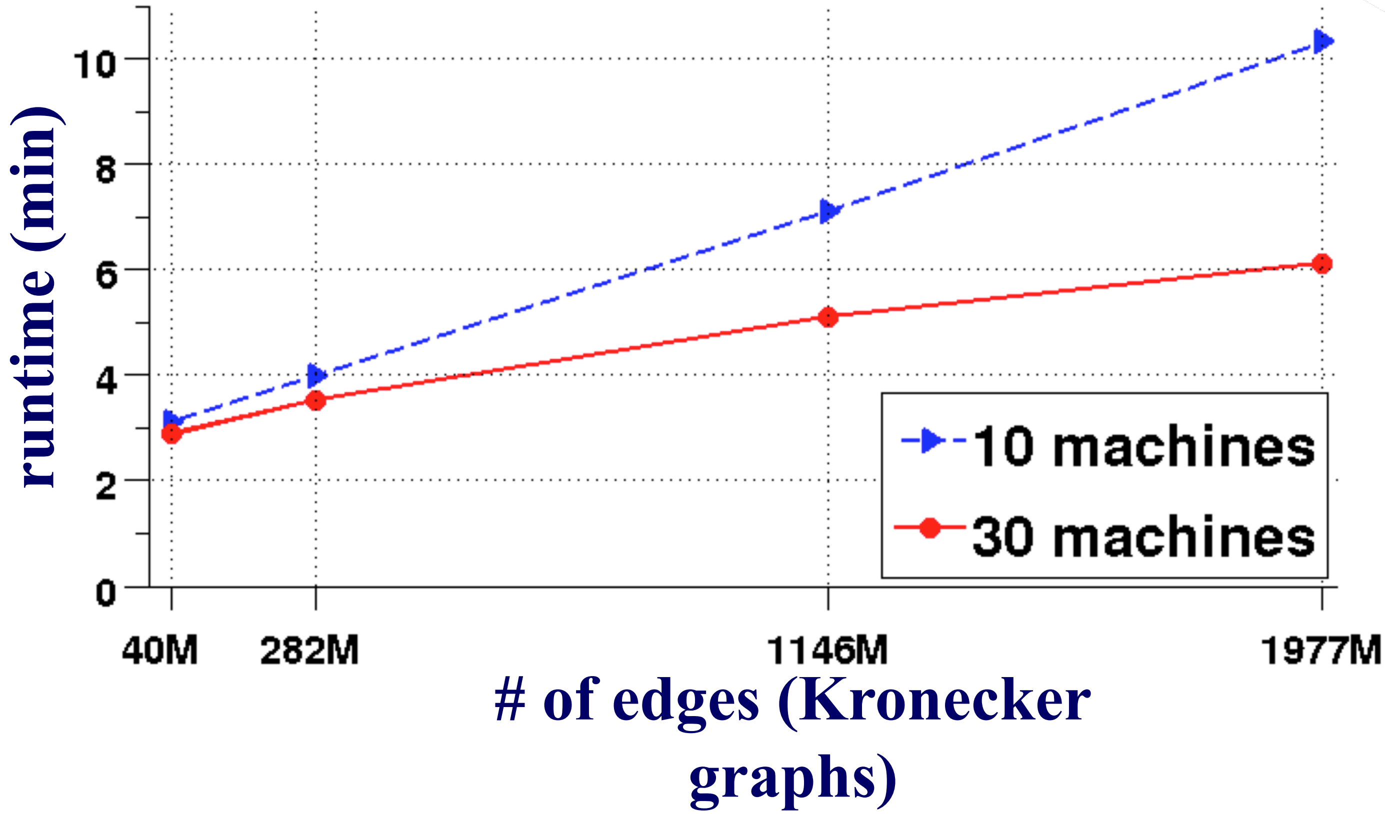
$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \quad \begin{bmatrix} d1 & & \\ & d2 & \\ & & d3 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} ? \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

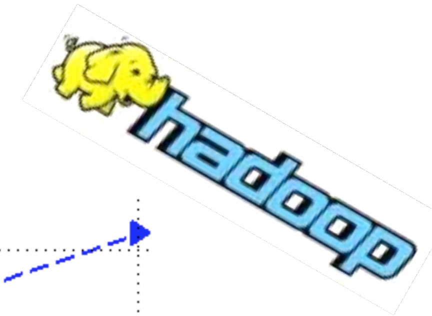
adjacency matrix

final labels/ beliefs

prior labels/ beliefs

Results: Scalability

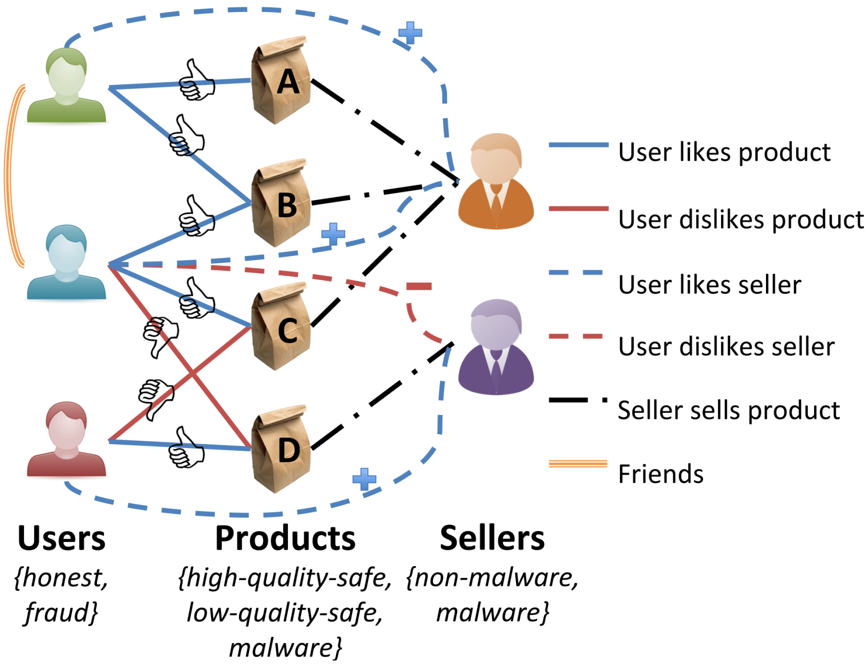


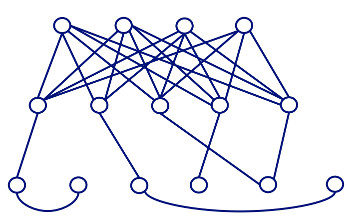


FABP is **linear** on the number of edges.

Carnegie Mellon

Problem: e-commerce ratings fraud

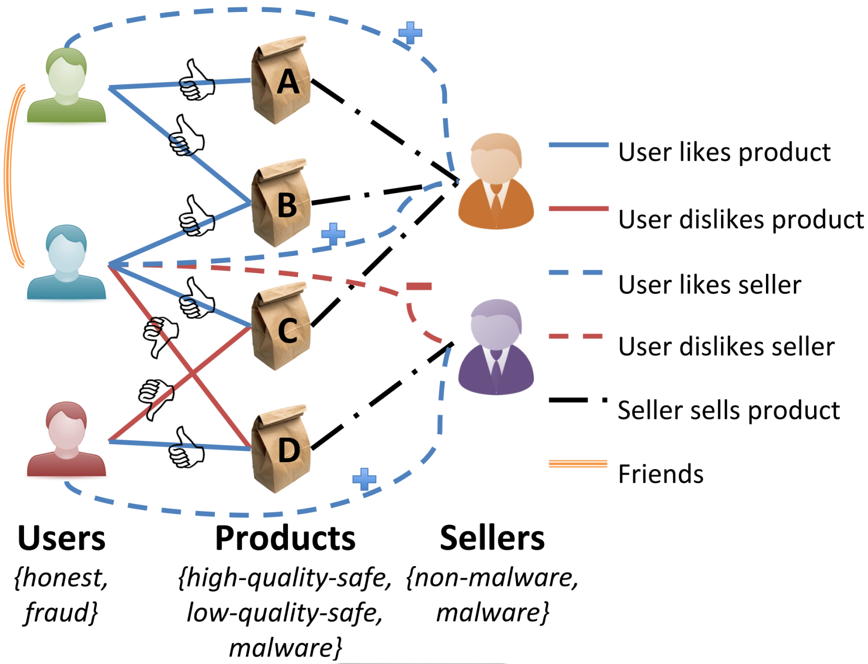




- **Given** a **heterogeneous** graph on users, products, sellers and positive/negative ratings with "seed labels"
- **Find** the top *k* most fraudulent users, products and sellers

Carnegie Mellon

Problem: e-commerce ratings fraud

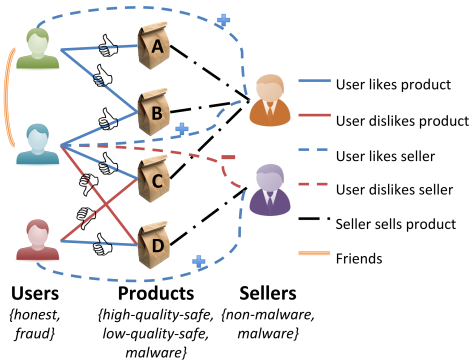


- **Given** a heterogeneous graph on users, products, sellers and positive/negative ratings with "seed labels"
- **Find** the top *k* most fraudulent users, products and sellers

<u>Dhivya Eswaran</u>, Stephan Günnemann, Christos Faloutsos, "*ZooBP:* Belief Propagation for Heterogeneous Networks", In submission to VLDB 2017

Carnegie Mellon

Problem: e-commerce ratings fraud



Theorem 1 (ZooBP). *If* $\mathbf{b}, \mathbf{e}, \mathbf{P}, \mathbf{Q}$ *are constructed as described above, the linear equation system approximating the final node beliefs given by BP is:*

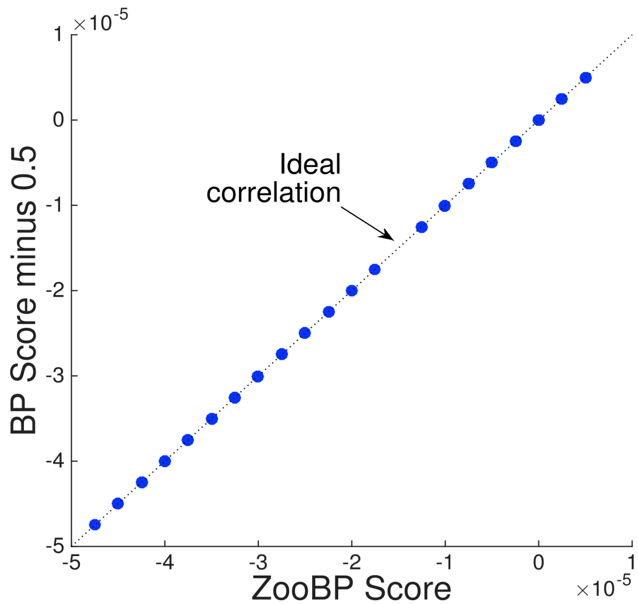
$$\mathbf{b} = \mathbf{e} + (\mathbf{P} - \mathbf{Q})\mathbf{b} \qquad (\textit{ZooBP}) \qquad (10)$$

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, "*ZooBP:* Belief Propagation for Heterogeneous Networks", In submission to VLDB 2017

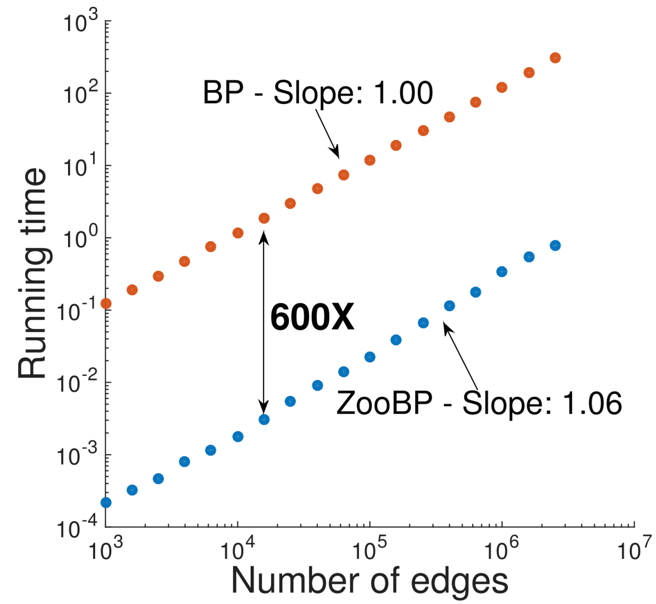
Carnegie Mellon

ZooBP: features

Fast; convergence guarantees.



Near-perfect accuracy

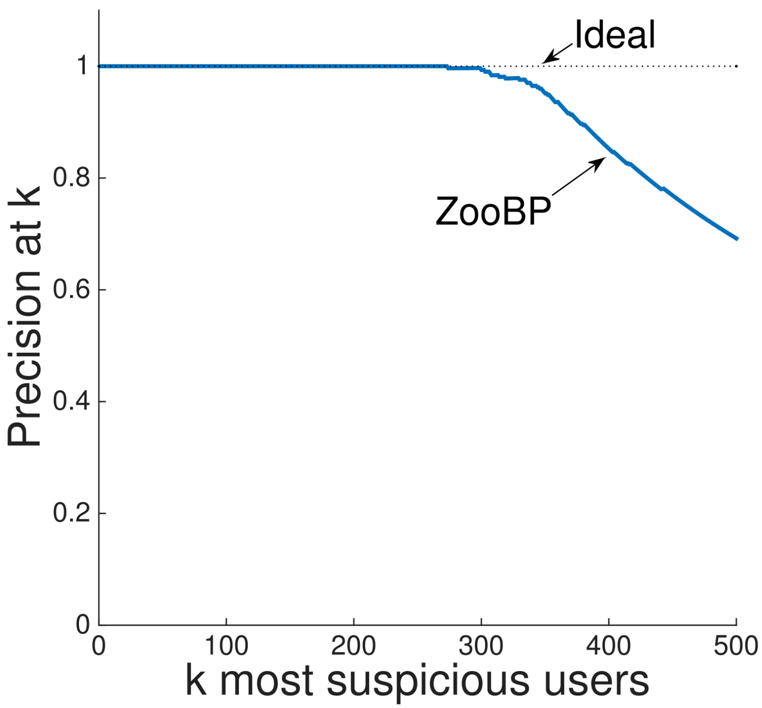


linear in graph size

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, "*ZooBP:* Belief Propagation for Heterogeneous Networks", In submission to VLDB 2017

Carnegie Mellon

ZooBP in the real world

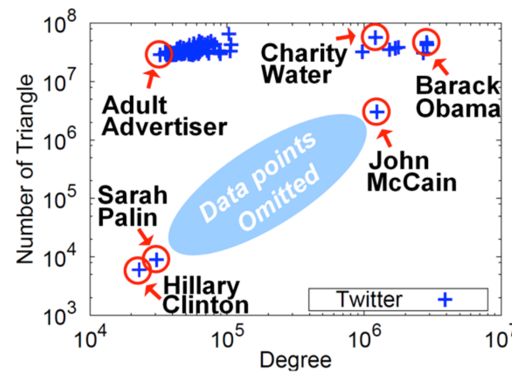


- Near 100% precision on top 300 users (Flipkart)
- Flagged users: suspicious
  - 400 ratings in 1 sec
  - 5000 good ratings and no bad ratings

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, "*ZooBP:* Belief Propagation for Heterogeneous Networks", In submission to VLDB 2017

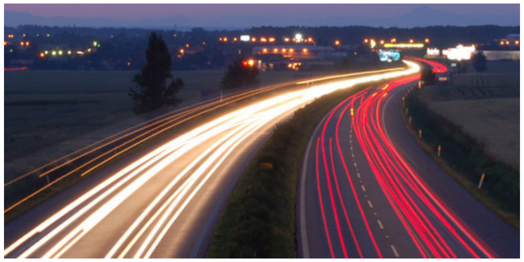
Summary of Part#1

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



Patterns  anomalies

Roadmap

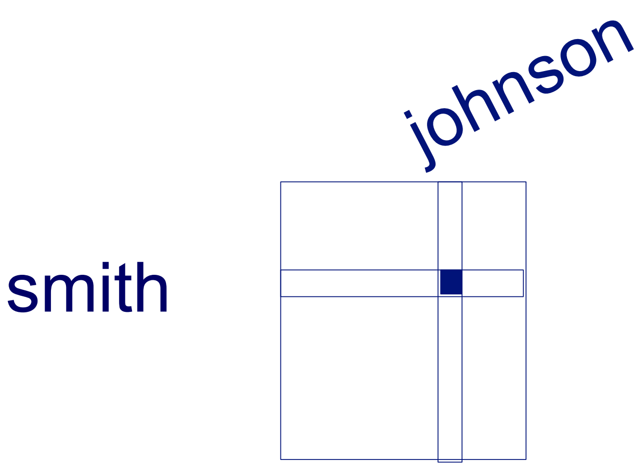


- Introduction – Motivation
- Part#1: Patterns in graphs
- Part#2: time-evolving graphs
  - P2.1: tools/tensors
  - P2.2: other patterns
- Conclusions

Part 2: Time evolving graphs; tensors

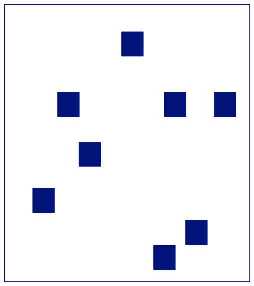
Graphs over time -> tensors!

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



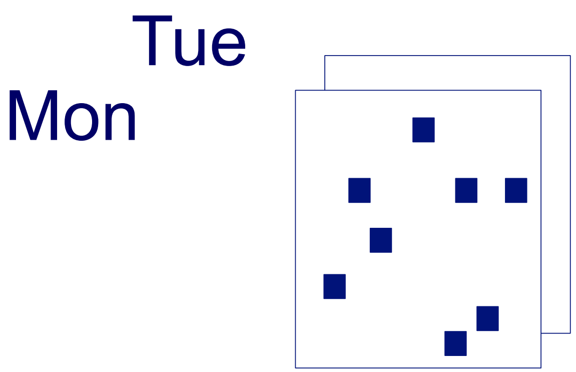
Graphs over time -> tensors!

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



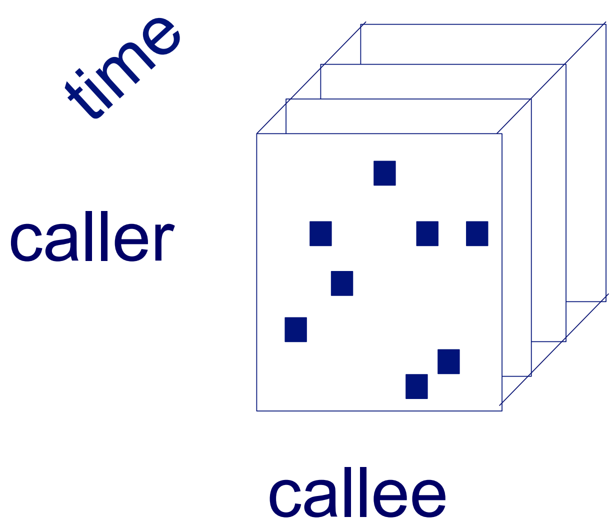
Graphs over time -> tensors!

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



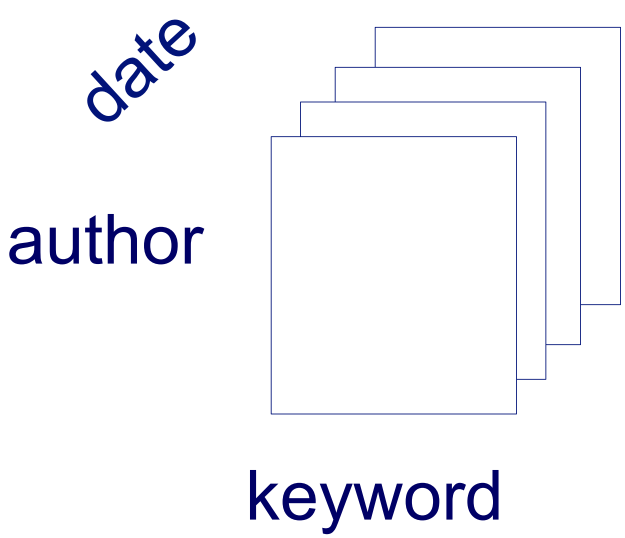
Graphs over time -> tensors!

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



Graphs over time -> tensors!

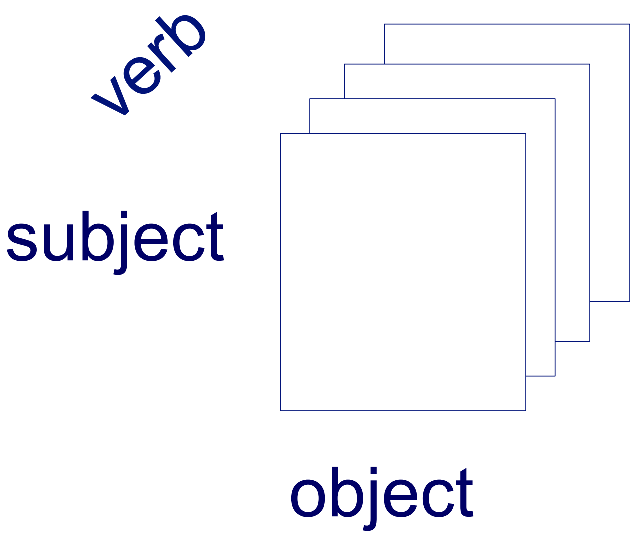
- Problem #2.1’:
  - Given author-keyword-date
  - Find patterns / anomalies



MANY more settings, with >2 ‘modes’

Graphs over time -> tensors!

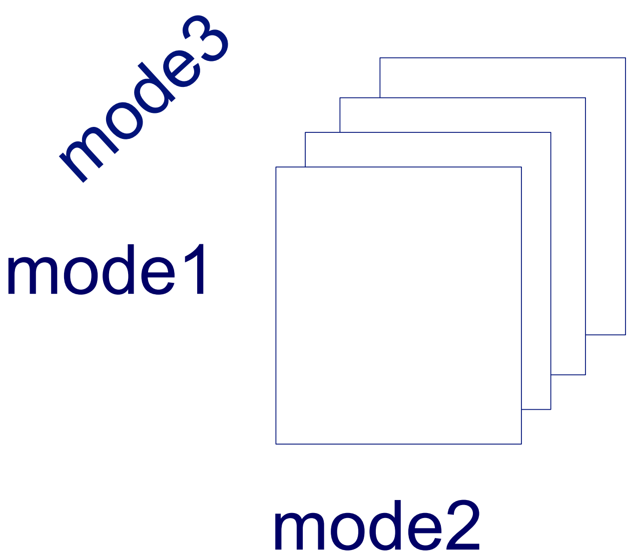
- Problem #2.1'':
  - Given subject – verb – object facts
  - Find patterns / anomalies



MANY more settings, with >2 'modes'

Graphs over time -> tensors!

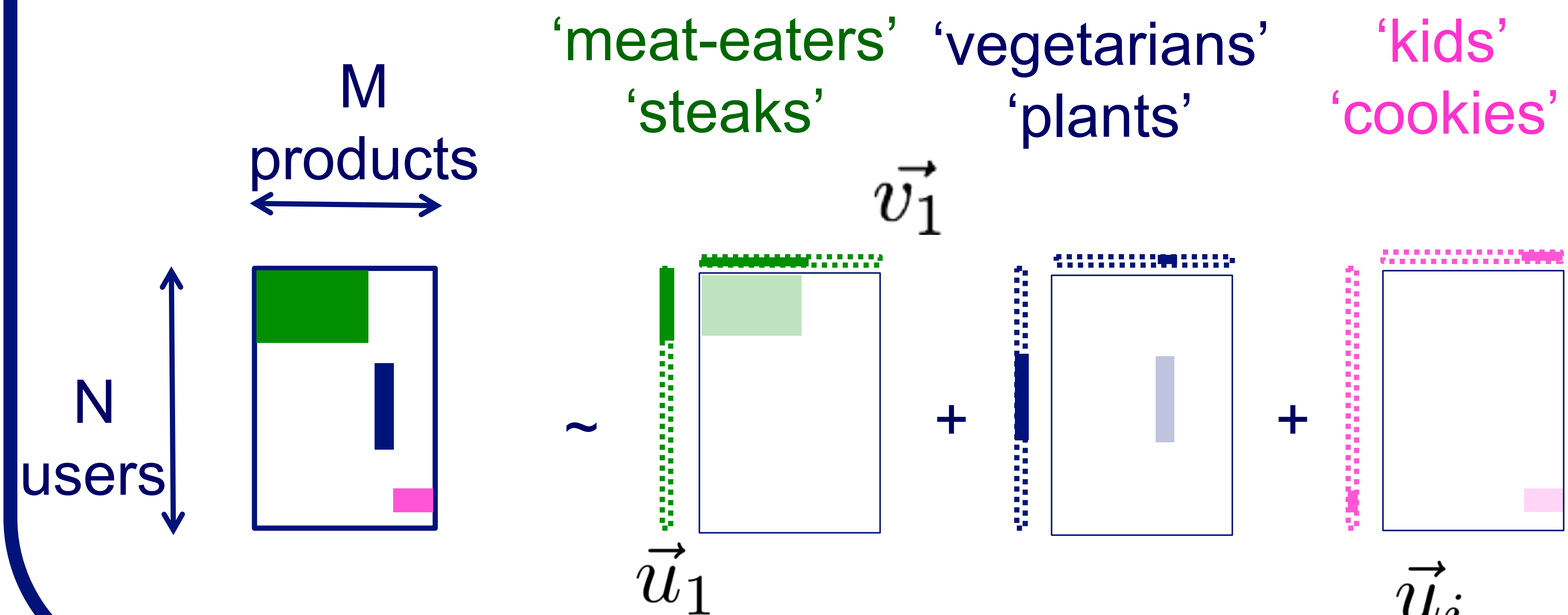
- Problem #2.1''':
  - Given <triplets>
  - Find patterns / anomalies



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

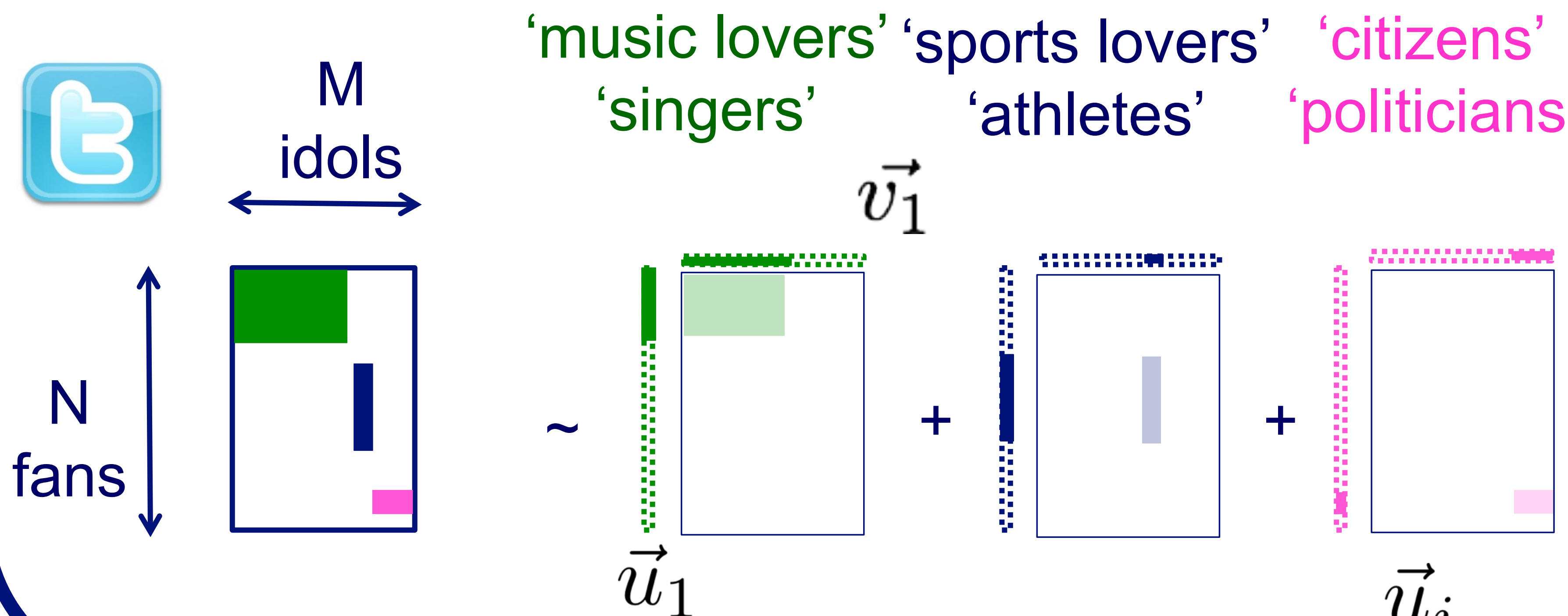
Answer : tensor factorization

- Recall: (SVD) matrix factorization: finds blocks



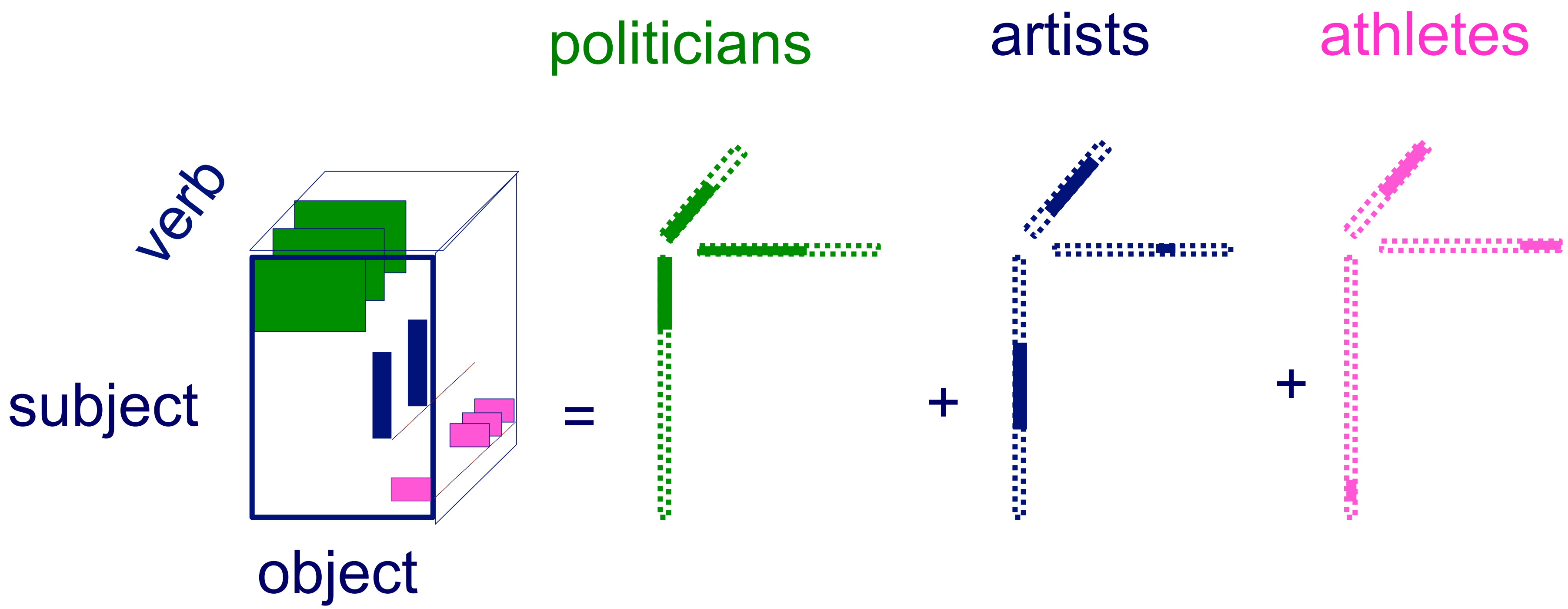
Crush intro to SVD

- Recall: (SVD) matrix factorization: finds blocks



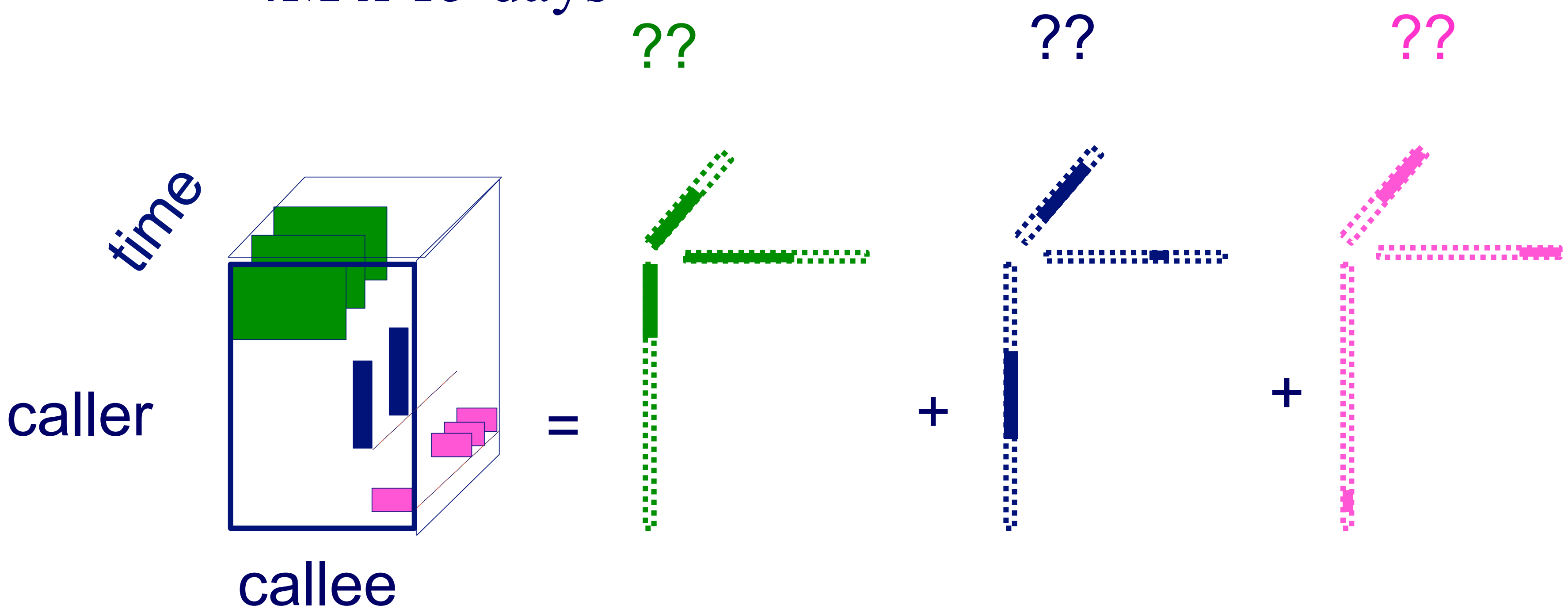
Answer: tensor factorization

- PARAFAC decomposition

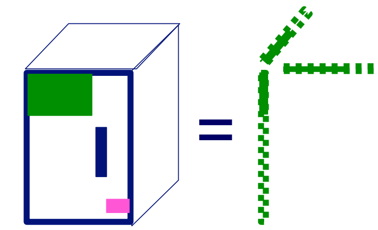


Answer: tensor factorization

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

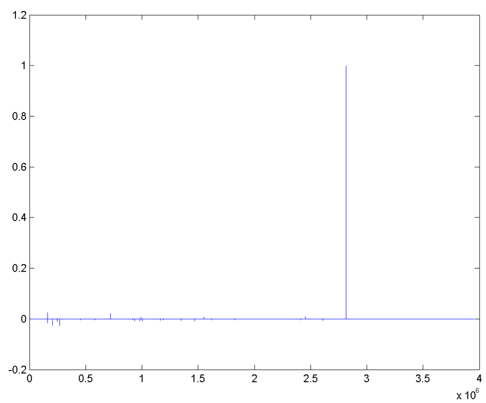


Anomaly detection in time-evolving graphs

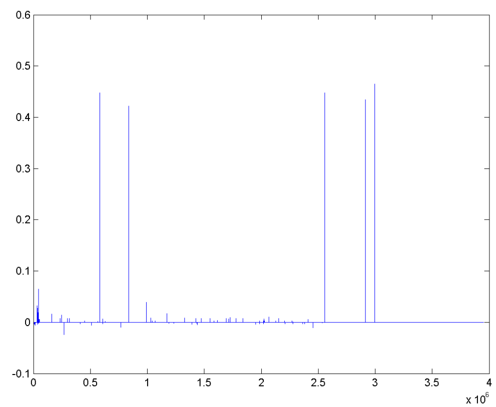


- Anomalous communities in phone call data:
  – European country, 4M clients, data over 2 weeks

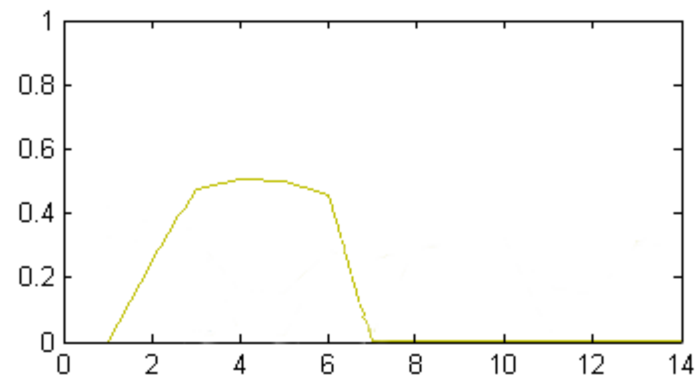
1 caller



5 receivers

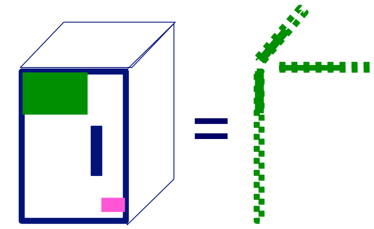


4 days of activity



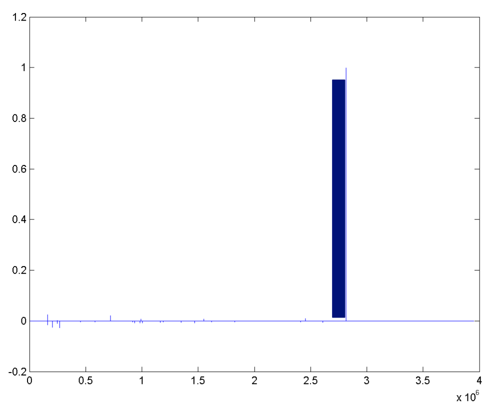
~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs

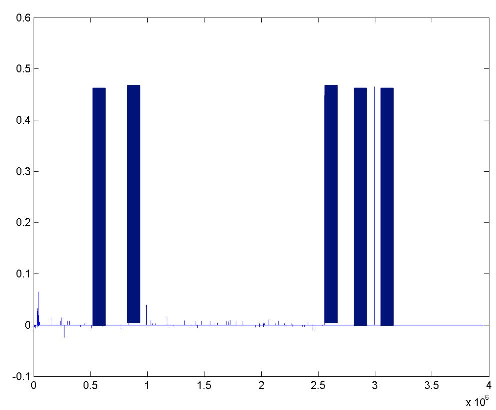


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

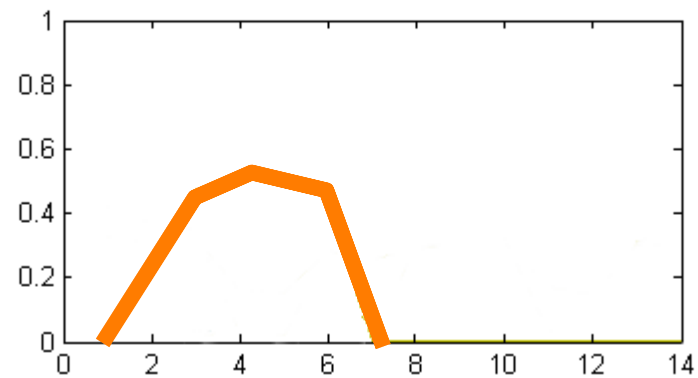
1 caller



5 receivers



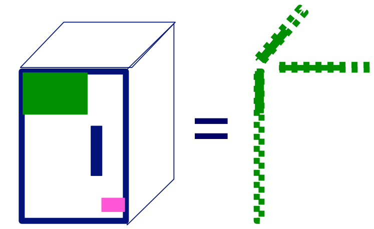
4 days of activity



~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs

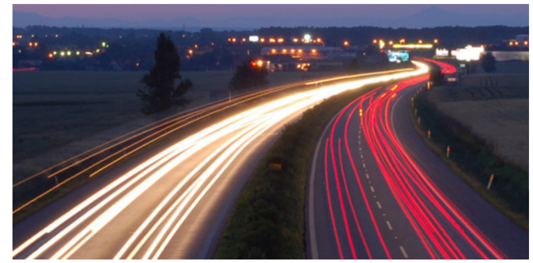


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

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.

Roadmap



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

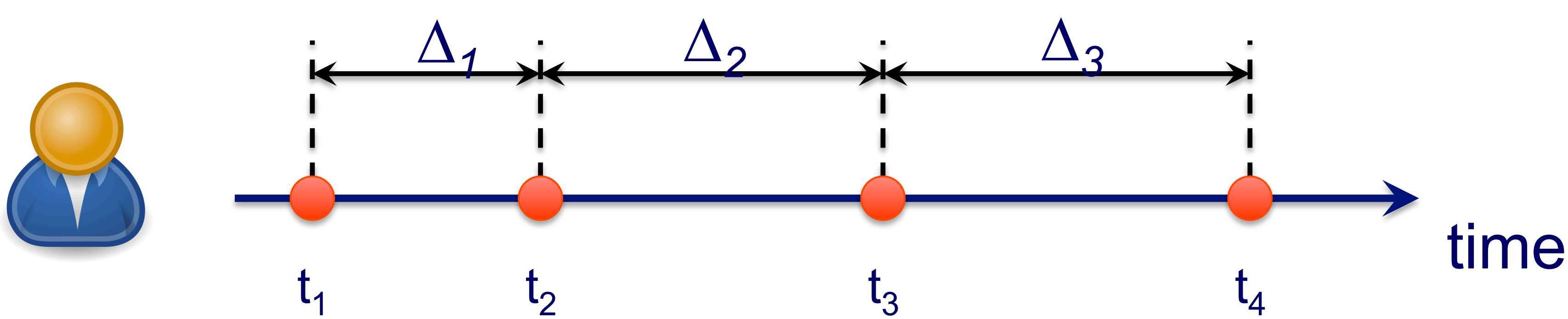
Pattern Mining: Datasets

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: $(\Delta_1, \Delta_2, \Delta_3, \ldots)$

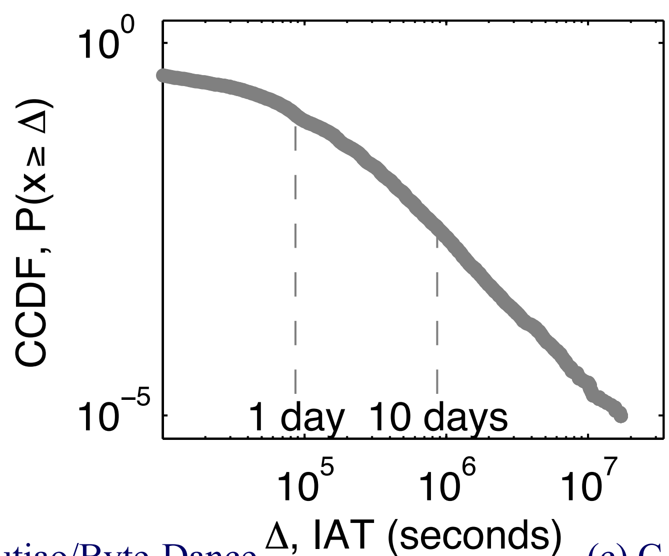


Pattern Mining

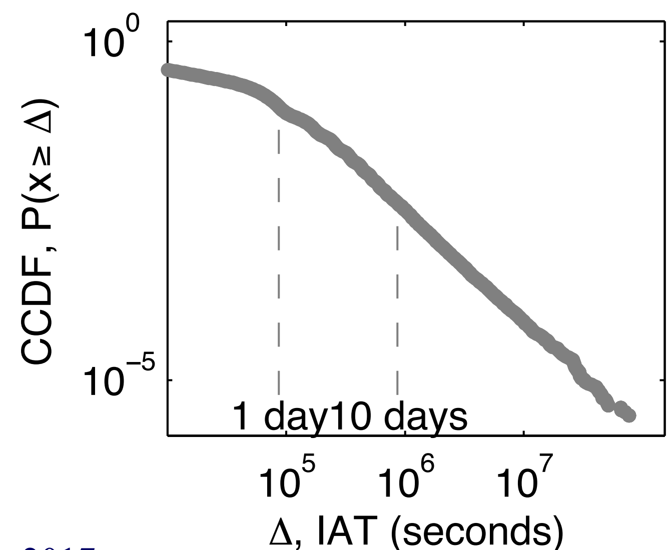
Pattern 1: Distribution of IAT is heavy-tailed

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

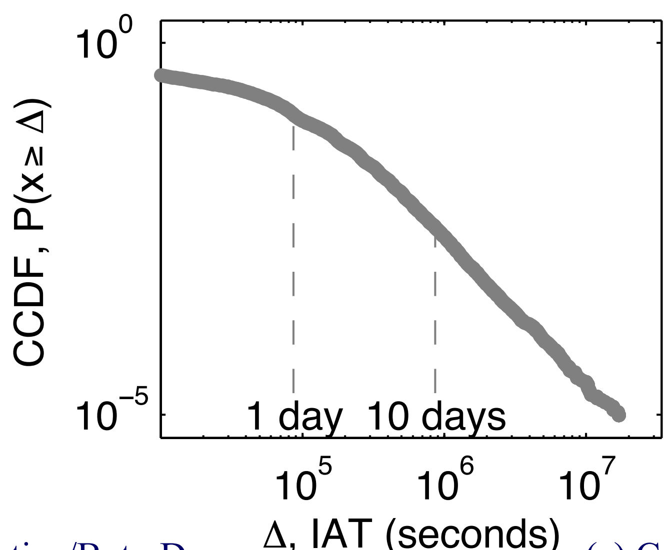
Pattern Mining

Pattern 1: Distribution of IAT is heavy-tailed

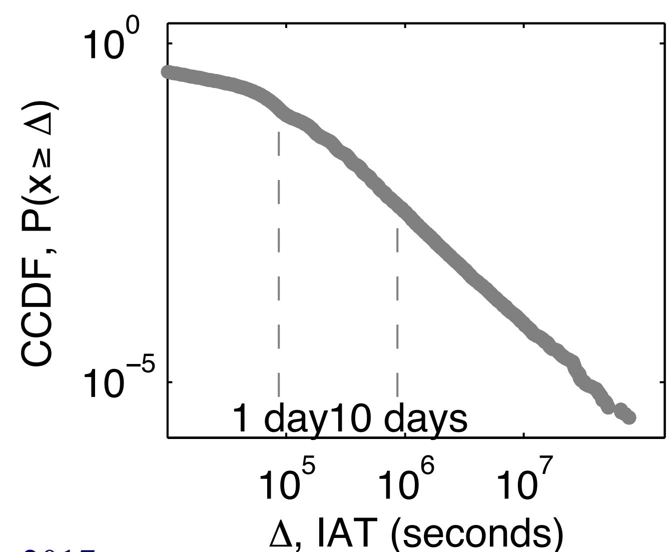
Users can be inactive for long periods of time before making new postings

No surprises – Should we give up?

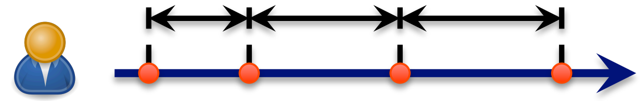
IAT Com … n (CCDF)



Reddit Users

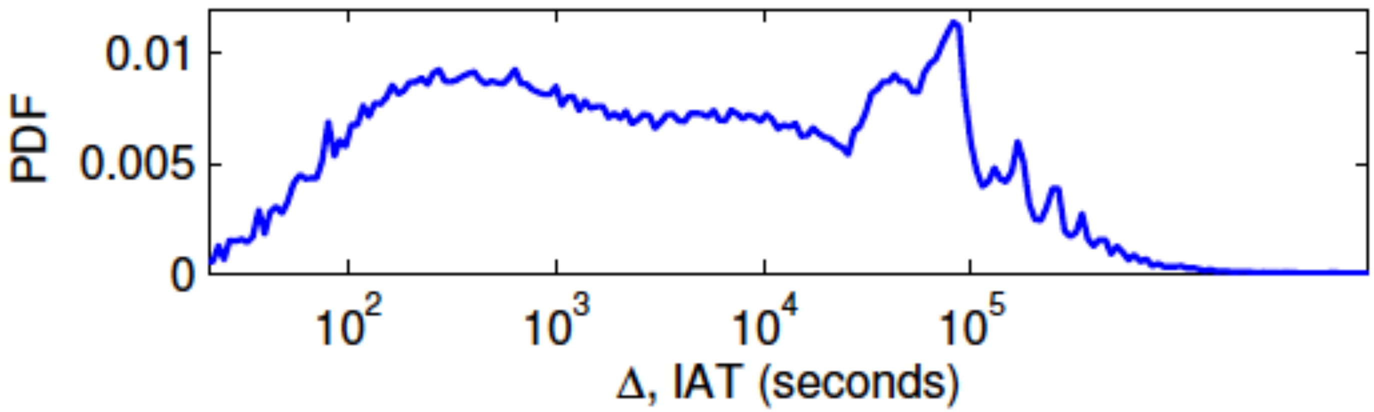


Twitter Users

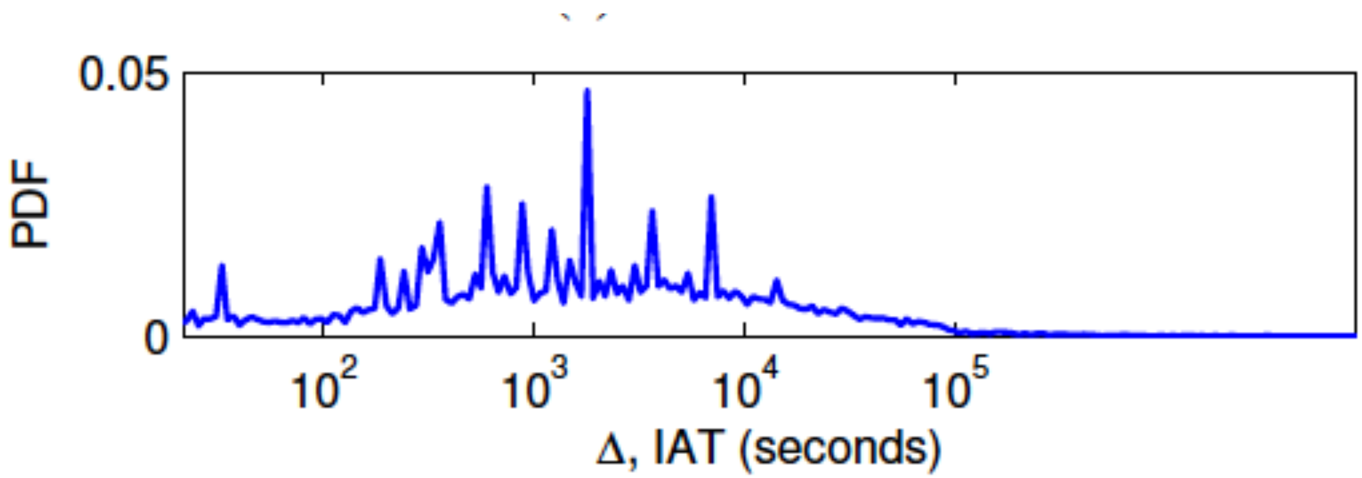


Human? Robots?

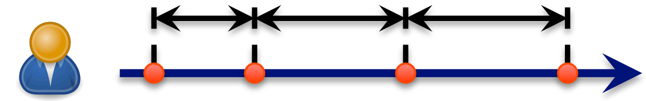
linear



log



Carnegie Mellon

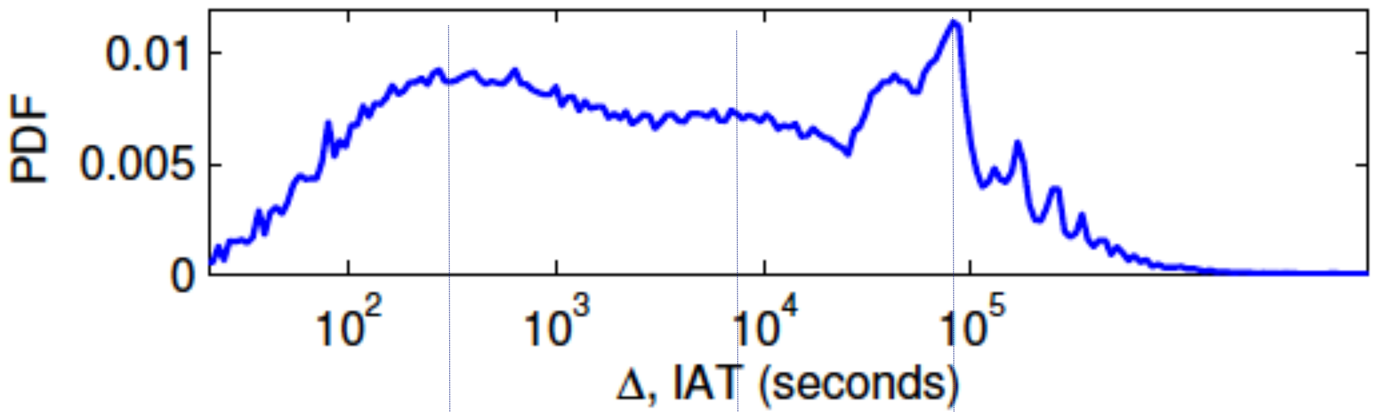


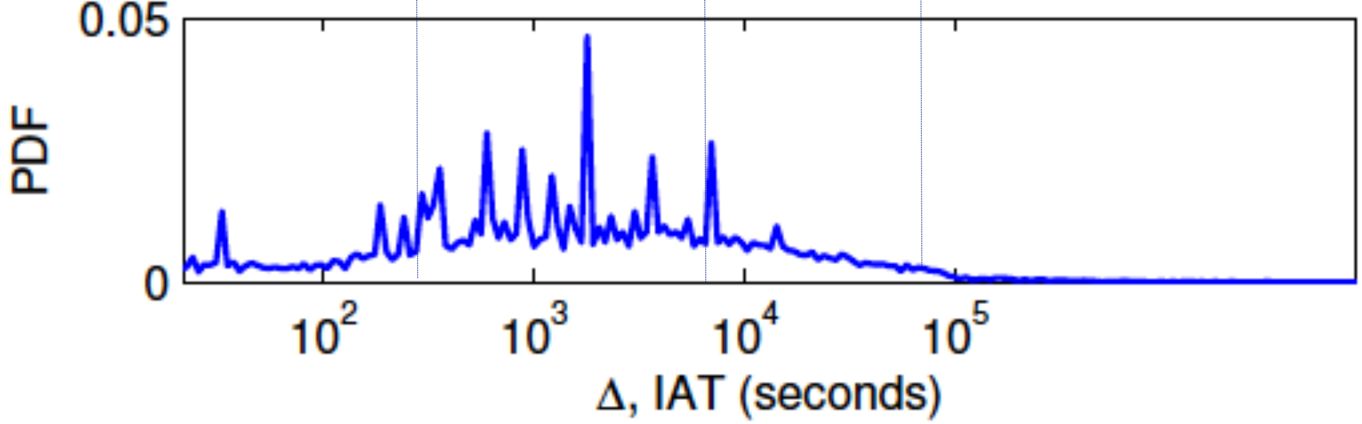
Human? Robots?

2’ 3h 1day

linear

log

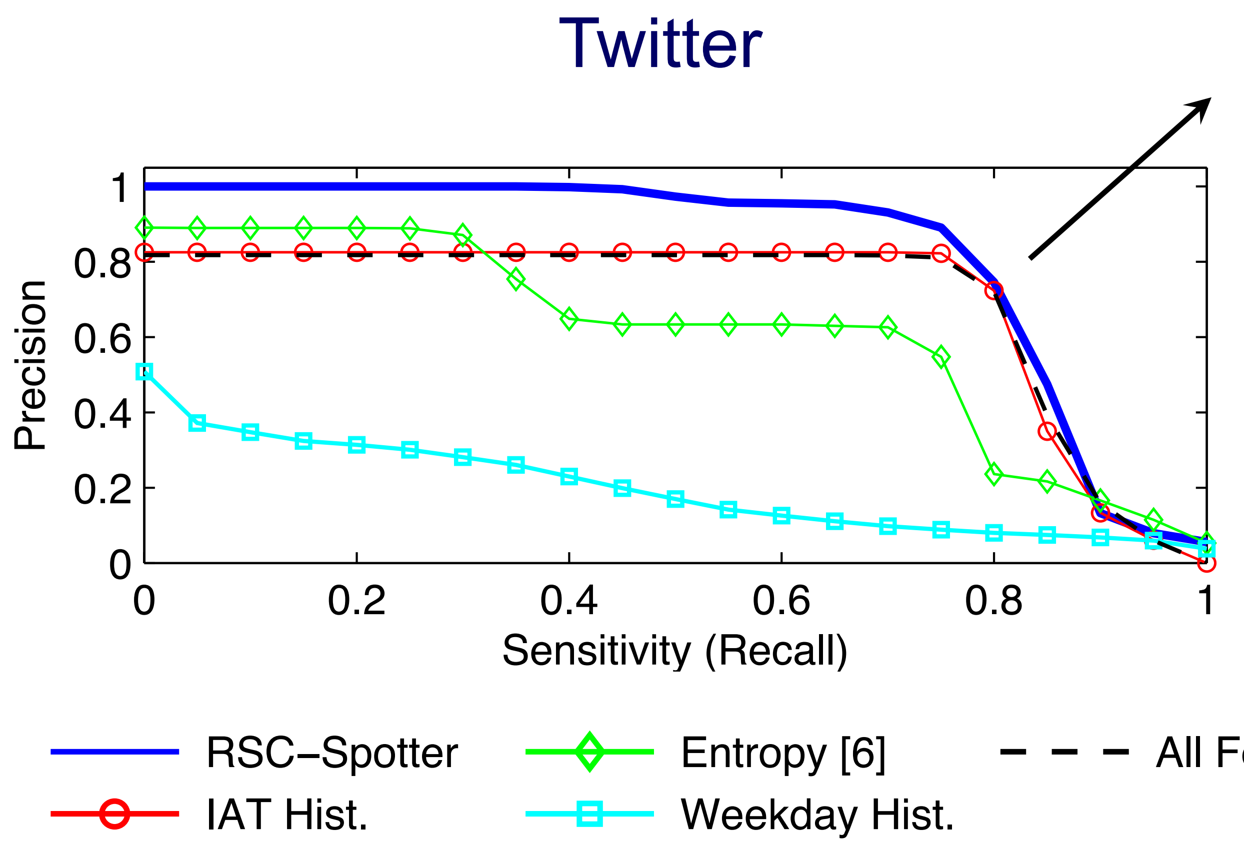




Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top



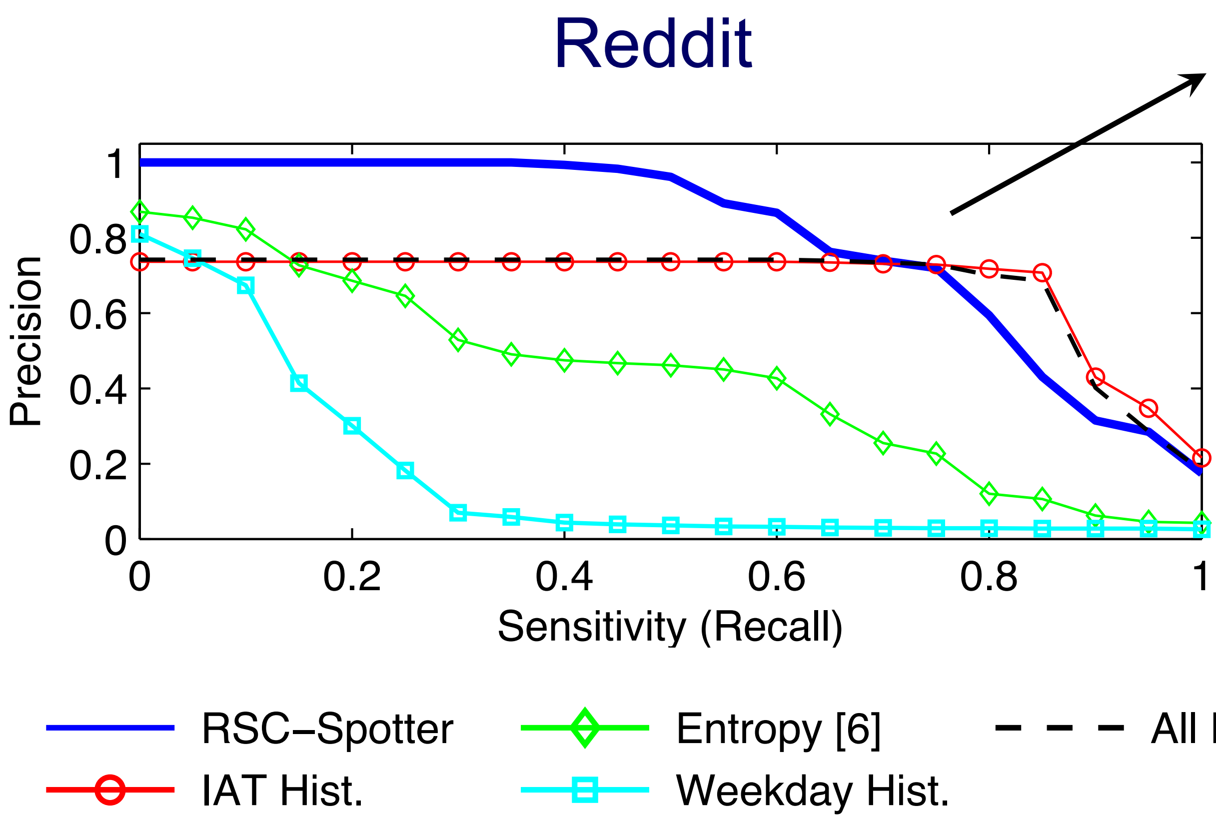
Precision > 94%
Sensitivity > 70%

With strongly imbalanced datasets
humans >> # bots

Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

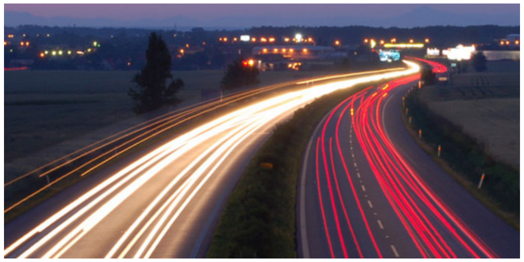
Good performance: curve close to the top



Precision > 96%
Sensitivity > 47%
With strongly imbalanced datasets
humans >> # bots

Carnegie Mellon

Roadmap



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

Carnegie Mellon

Beyond Sigmoids: the NetTide Model for Social Network Growth and its Applications

KDD'16

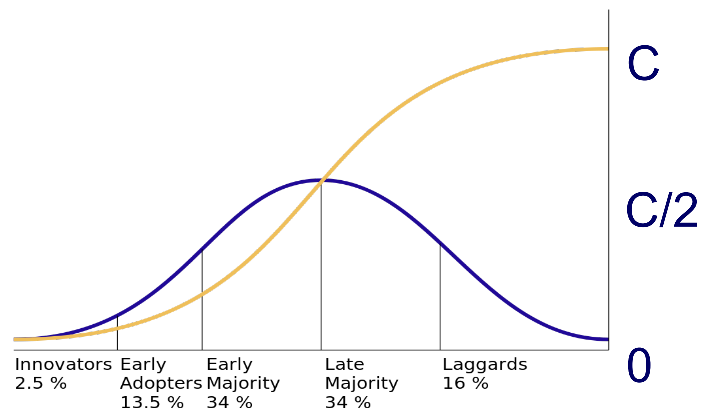
Chengxi Zang 臧承熙, Peng Cui, CF

Carnegie Mellon

PROBLEM: n(t) and e(t), over time?

- n(t): the number of nodes.
- e(t): the number of edges.
- E.g.:
  - How many members will have next month?
  - How many friendship links will have next year?

- Linear?
- Exponential?
- Sigmoid?

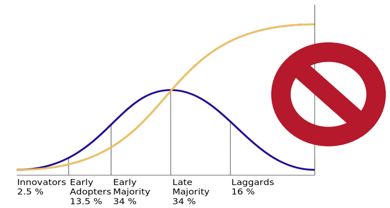


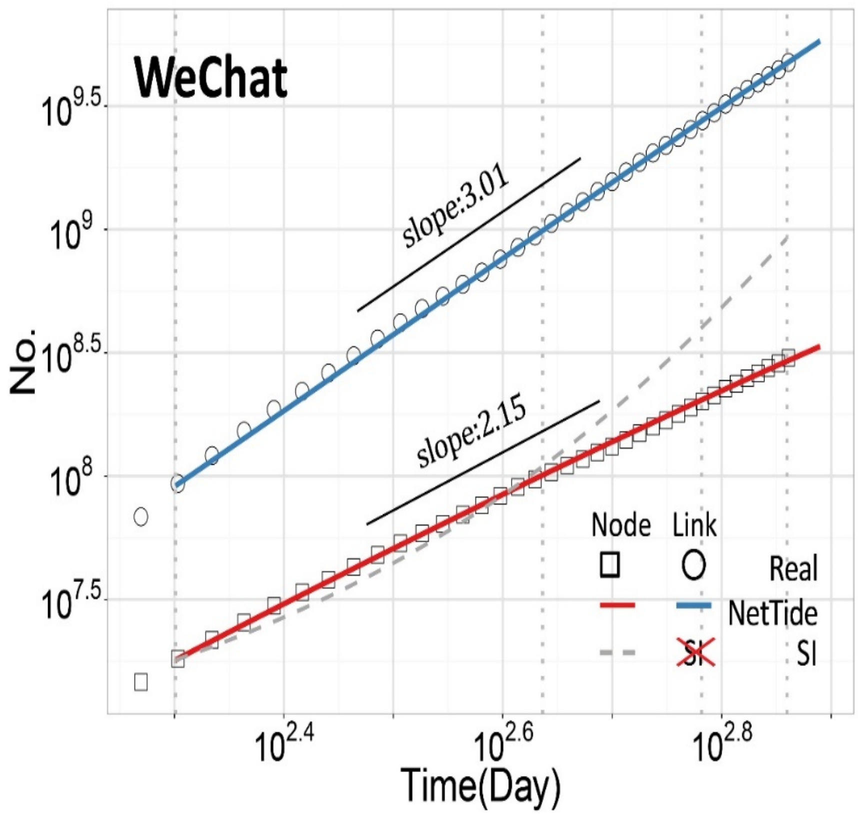
Datasets

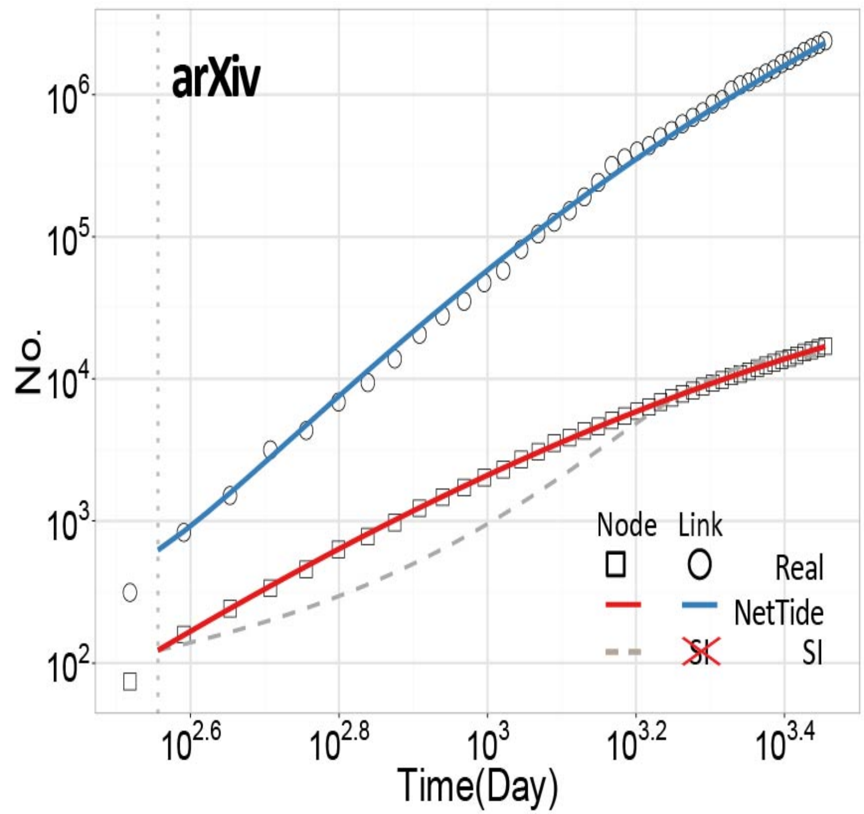
- **WeChat 2011/1-2013/1 300M nodes, 4.75B links**
- ArXiv 1992/3-2002/3 17k nodes, 2.4M links
- Enron 1998/1-2002/7 86K nodes, 600K links
- Weibo 2006 165K nodes, 331K links

Carnegie Mellon

A: Power Law Growth







Cumulative growth（Log-Log scale）

Details

Proposed: NetTide Model

- Nodes n(t)

$$\frac{dn(t)}{dt} = \frac{\beta}{t^{\theta}} n(t)(N - n(t))$$

- Links e(t)

$$\frac{de(t)}{dt} = \frac{\beta'}{t^{\theta}} n(t) \left(\alpha (n(t) - 1)^{\gamma} - \frac{e(t)}{n(t)} \right) + 2 \frac{dn(t)}{dt}$$

Carnegie Mellon

Details

NetTide-Node Model

$$\frac{dn(t)}{dt} = \frac{\beta}{\boxed{t^{\theta}}} n(t) \; (N - n(t))$$

#nodes(t)

Total population

- Intuition:
  - **Rich-get-richer**
  - Limitation
  - Fizzling nature

= SI; ~Bass

Carnegie Mellon

Details

NetTide-Node Model

$$\frac{dn(t)}{dt} = \frac{\beta}{\boxed{t^{\theta}}} n(t) \left(N - n(t)\right)$$

#nodes(t)

Total population

- Intuition:
  - Rich-get-richer
  - **Limitation**
  - Fizzling nature

= SI; ~Bass

Carnegie Mellon

Details

NetTide-Node Model

$$\frac{dn(t)}{dt} = \frac{\beta}{t^{\theta}} \, n(t) \, (N - n(t))$$

#nodes(t)

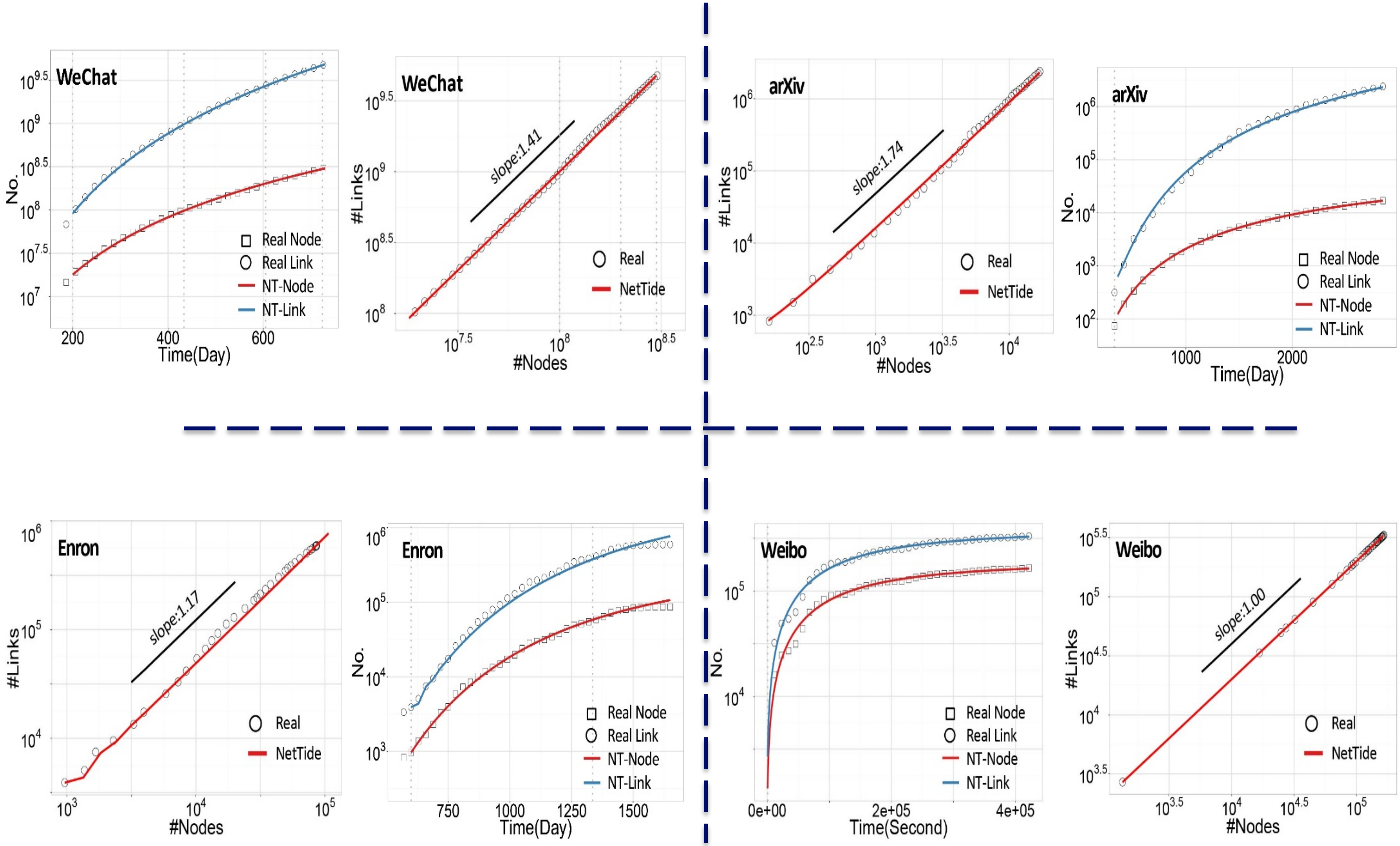
Total population

- Intuition:
  - Rich-get-richer
  - Limitation
  - **Fizzling nature**

= SI; ~Bass

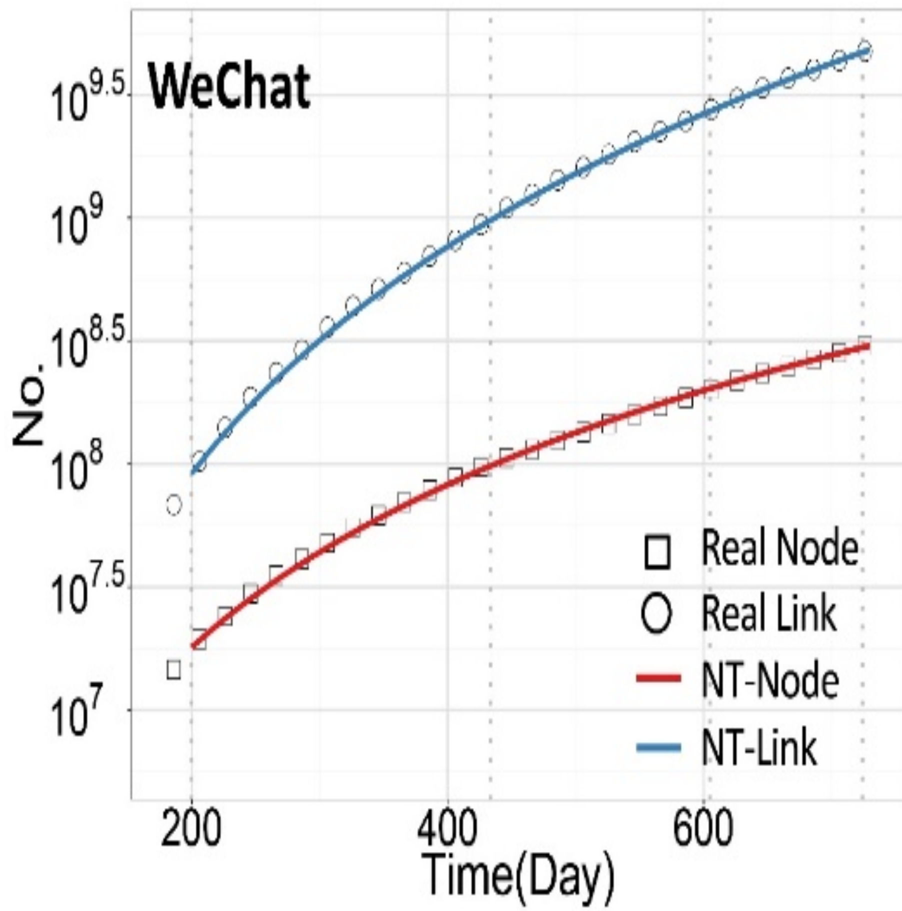
Carnegie Mellon

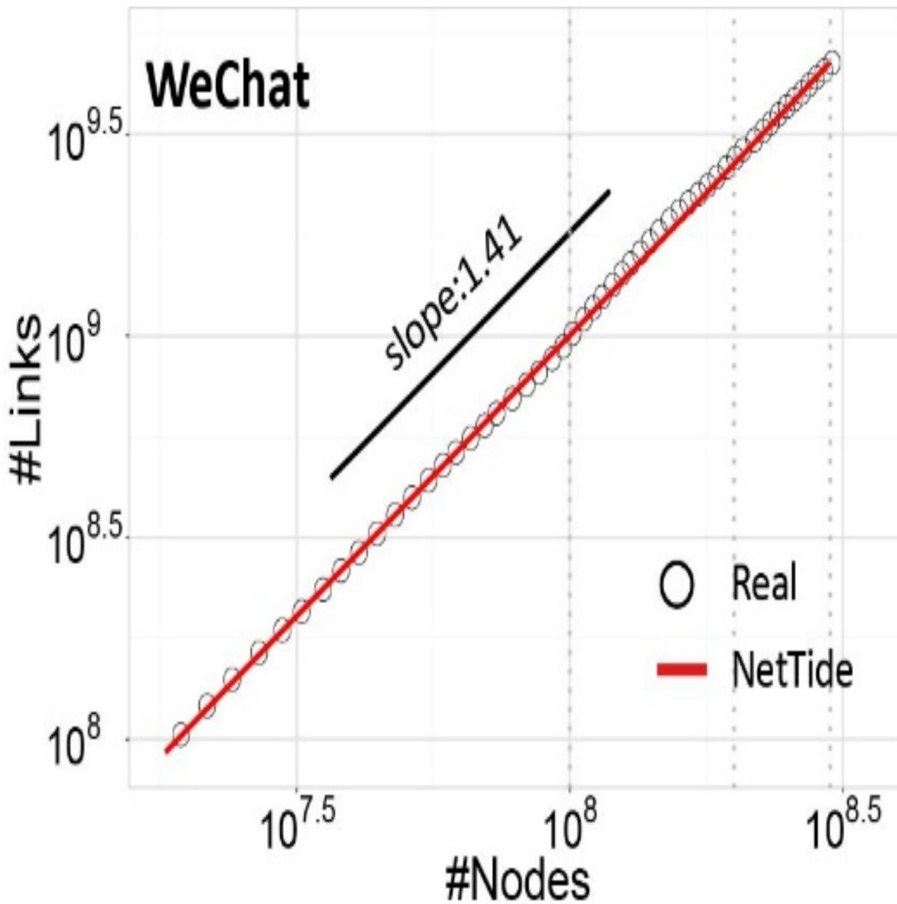
Results: Accuracy



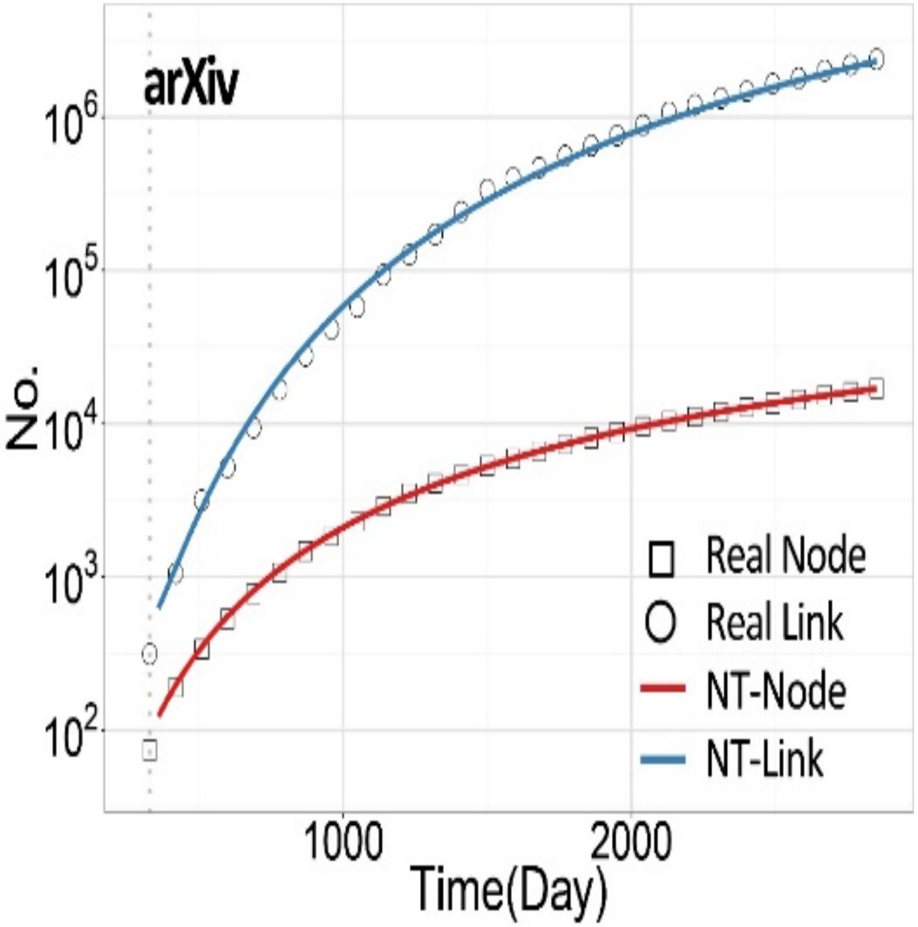
Carnegie Mellon

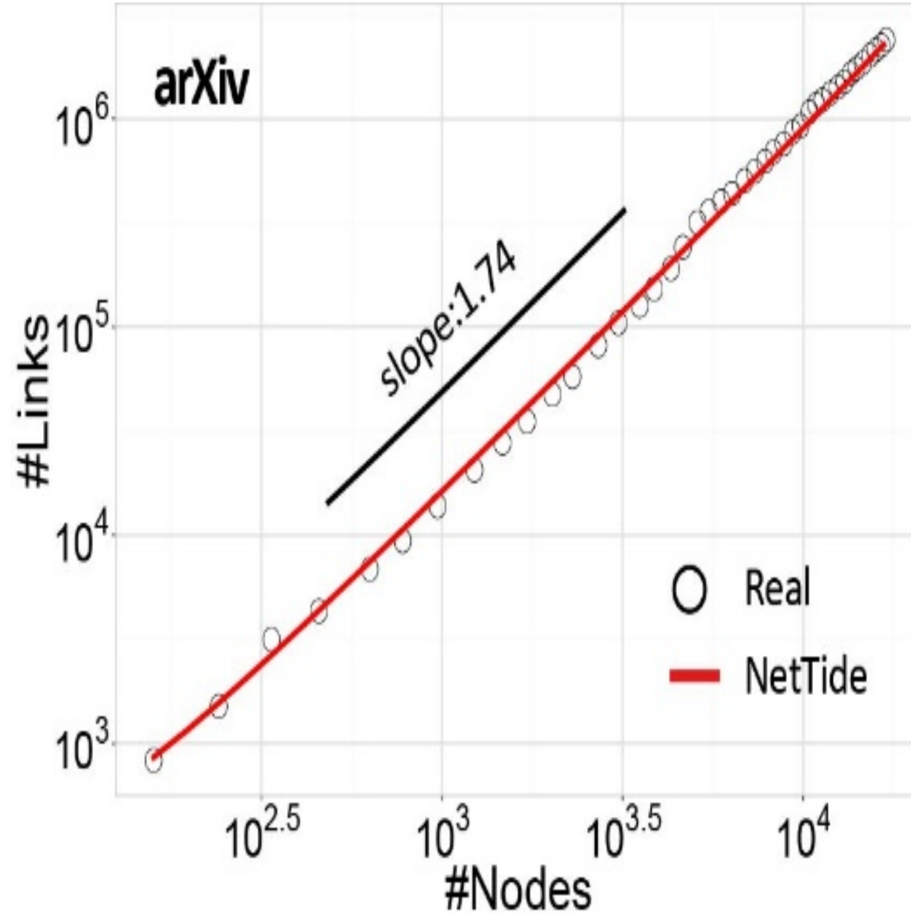
Results: Accuracy





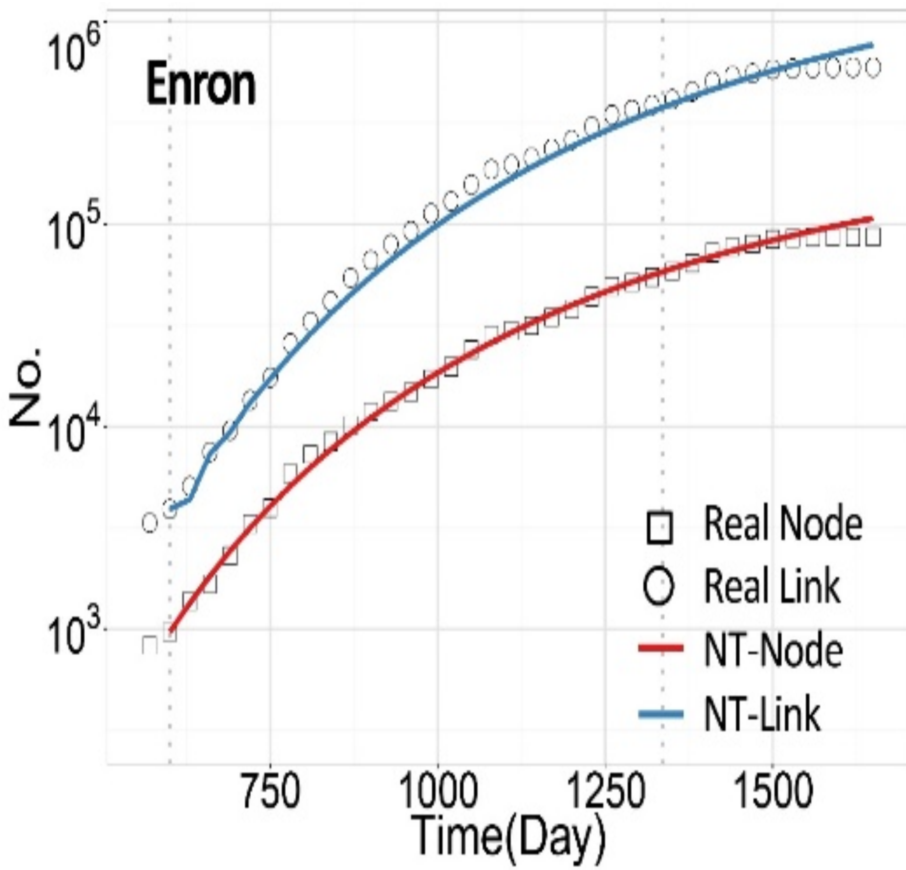
Results: Accuracy

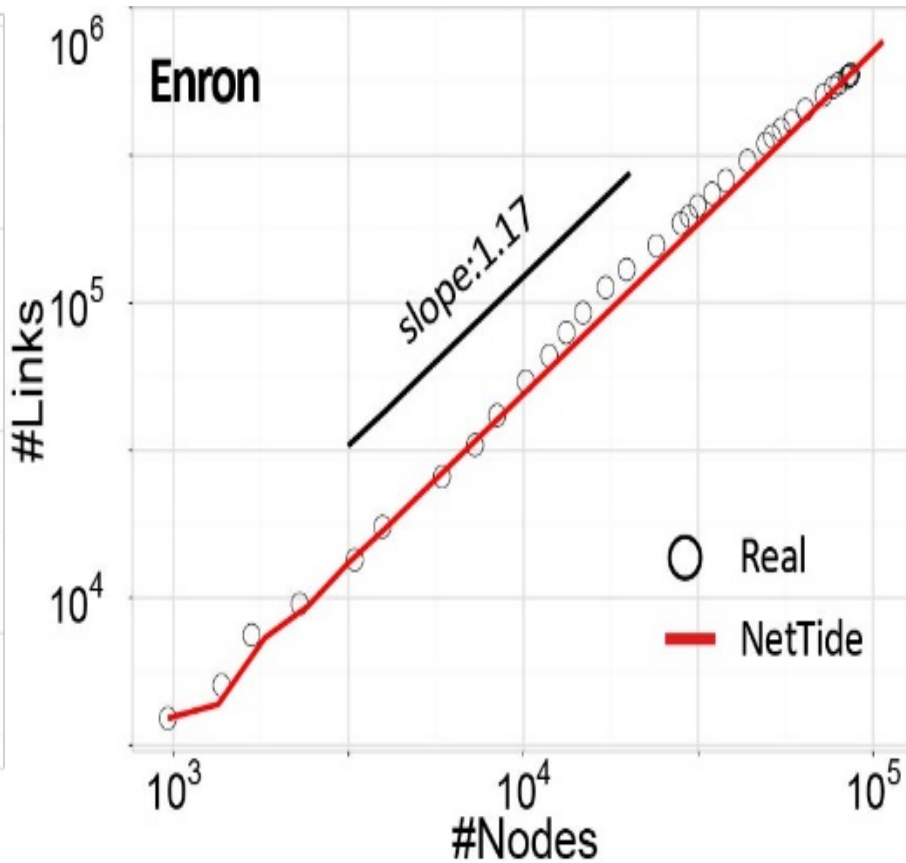




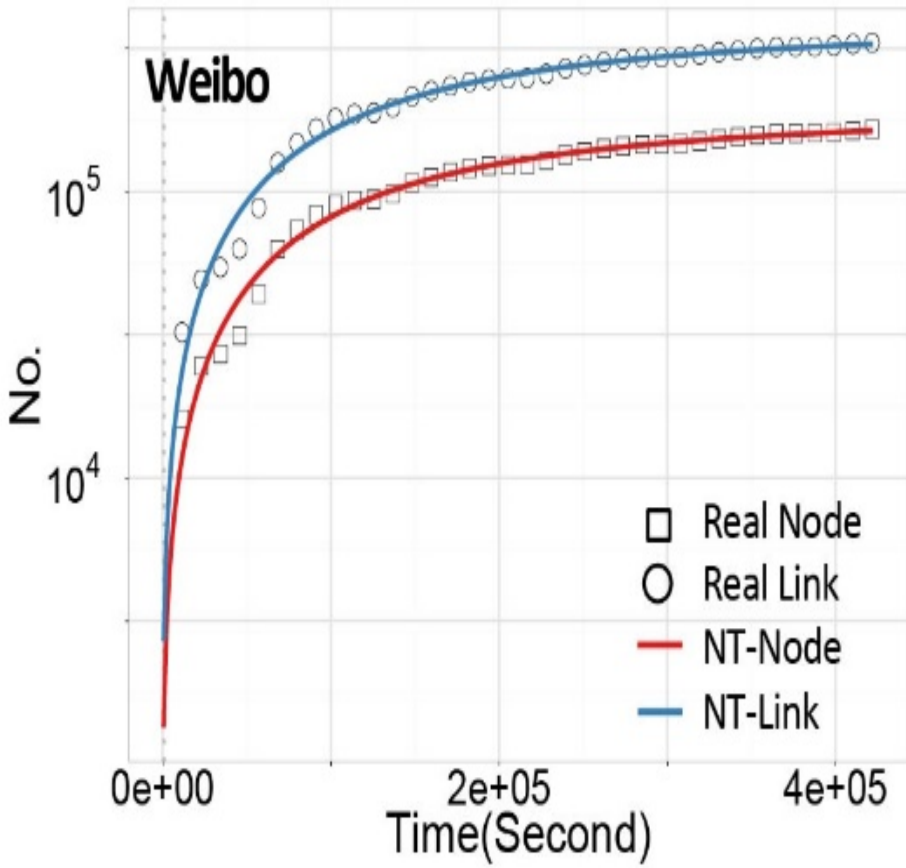
Carnegie Mellon

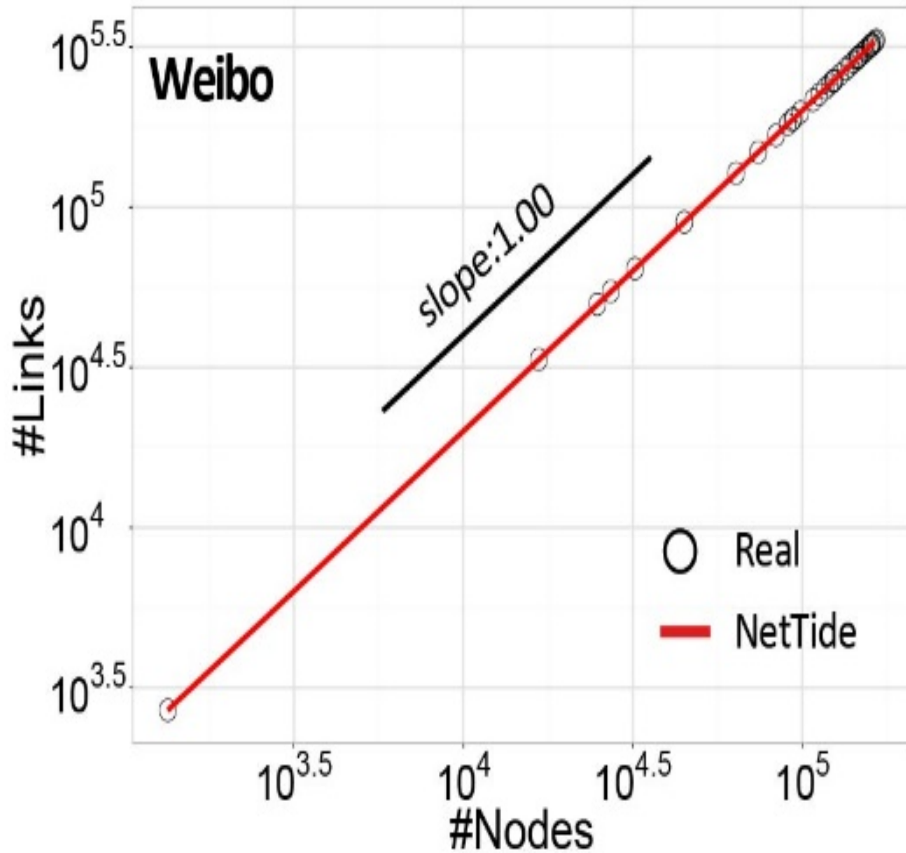
Results: Accuracy





Results: Accuracy



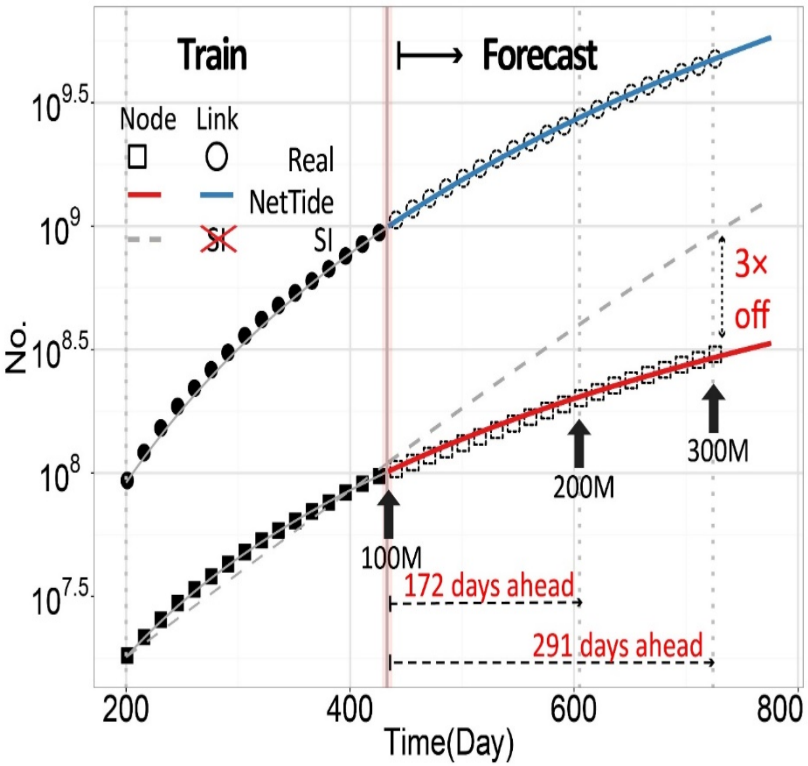


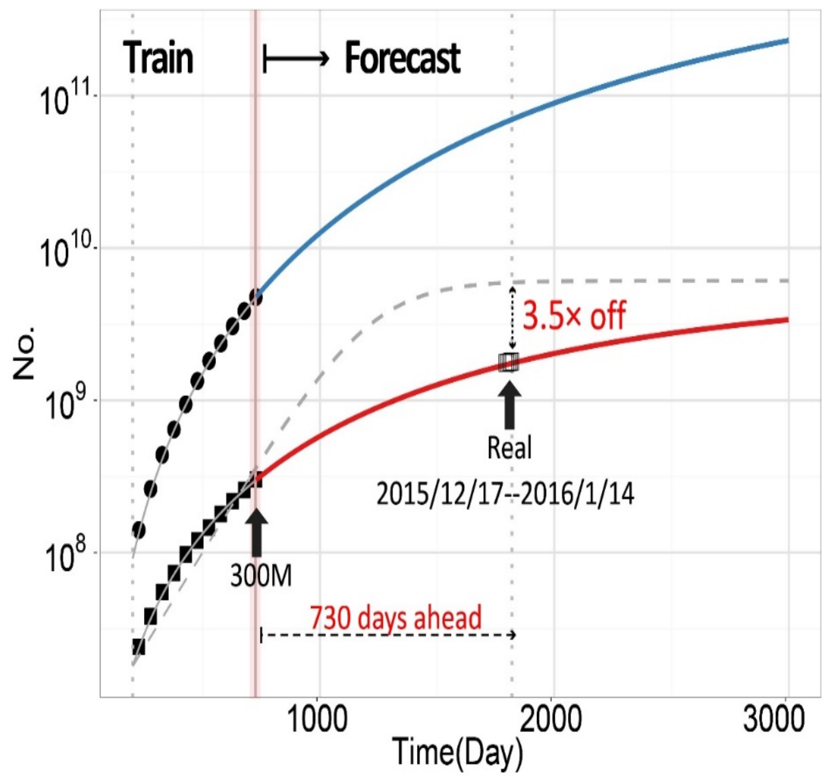
Carnegie Mellon

Results: Forecast

WeChat from 100 million to 300 million

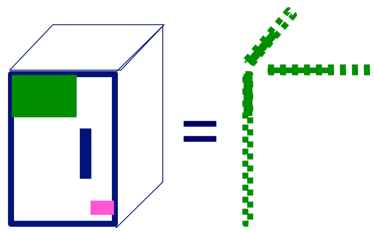
730 days ahead

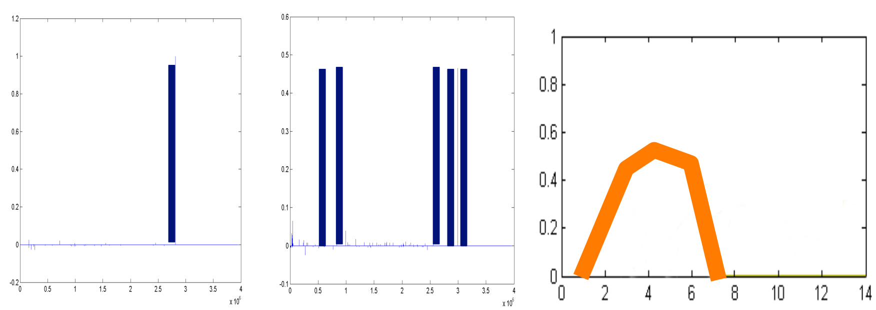




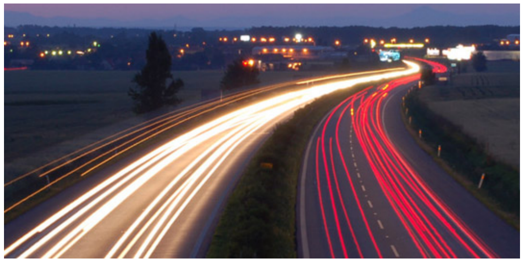
Part 2: Conclusions

- Time-evolving / heterogeneous graphs -> tensors
- PARAFAC finds patterns
- Surprising temporal patterns (P.L. growth)





Roadmap



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

Carnegie Mellon

Thanks

Disclaimer: All opinions are mine; not necessarily reflecting the opinions of the funding agencies

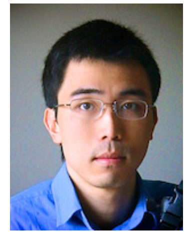
Thanks to: NSF IIS-0705359, IIS-0534205, CTA-INARC; Yahoo (M45), LLNL, IBM, SPRINT, Google, INTEL, HP, iLab

Cast

Akoglu, Leman

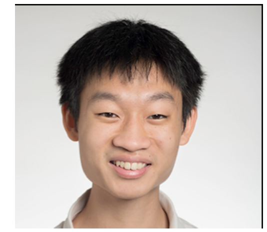
Araujo, Miguel

Beutel, Alex



Chau, Polo

Eswaran, Dhivya

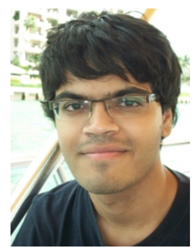


Hooi, Bryan

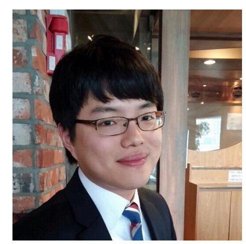
Kang, U

Koutra, Danai

Papalexakis, Vagelis



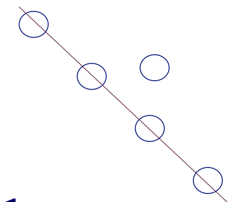
Shah, Neil



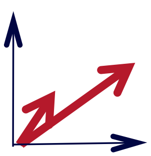
Shin, Kijung

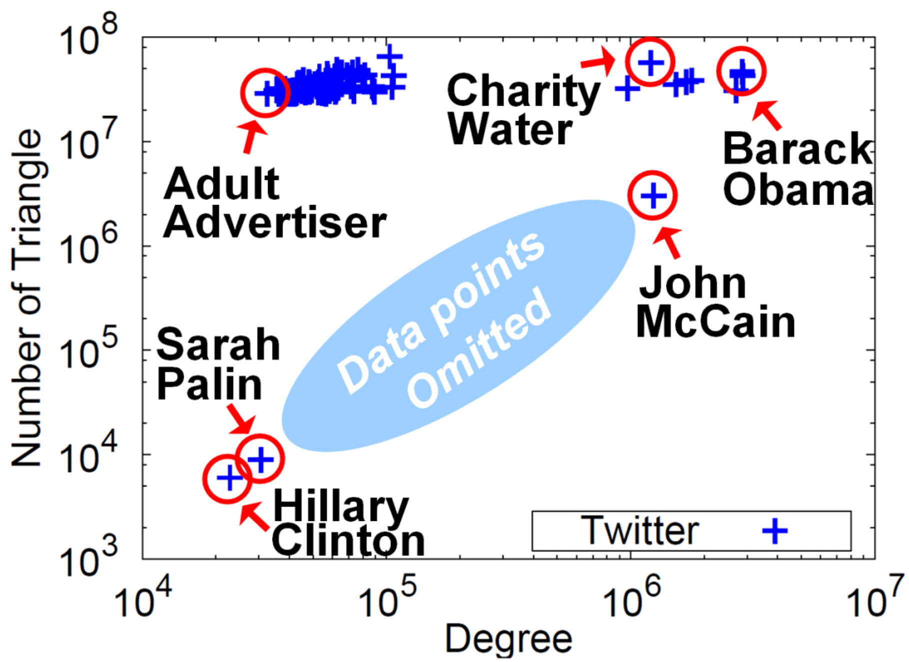
Song, Hyun Ah

CONCLUSION#1 – Big data



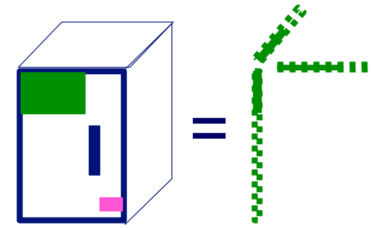
- **Patterns**  **Anomalies**
- **Large** datasets reveal patterns/outliers that are invisible otherwise

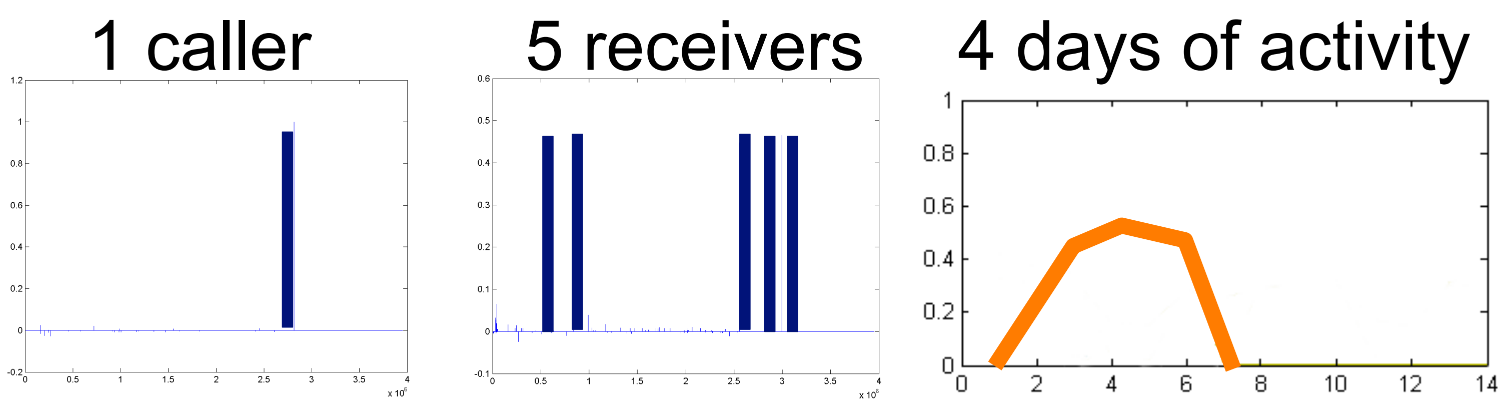




CONCLUSION#2 – tensors

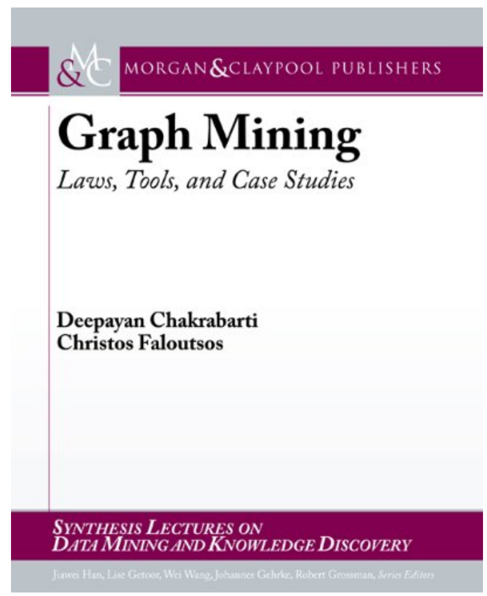
- powerful tool





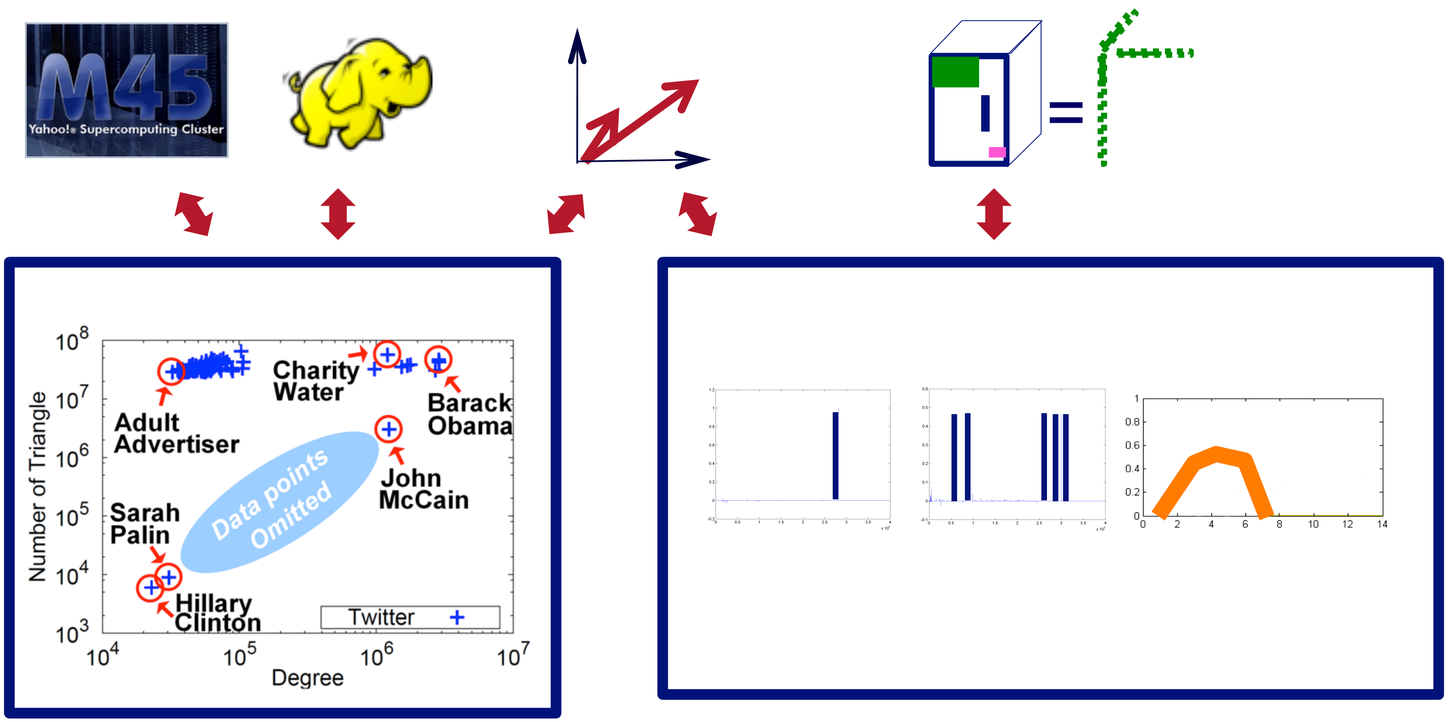
References

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



TAKE HOME MESSAGE:

Cross-disciplinarity



Carnegie Mellon

Thank you!

Cross-disciplinarity