Carnegie Mellon

15-826: Multimedia (Databases) and Data Mining

Anomaly detection in large graphs

Christos Faloutsos

Roadmap

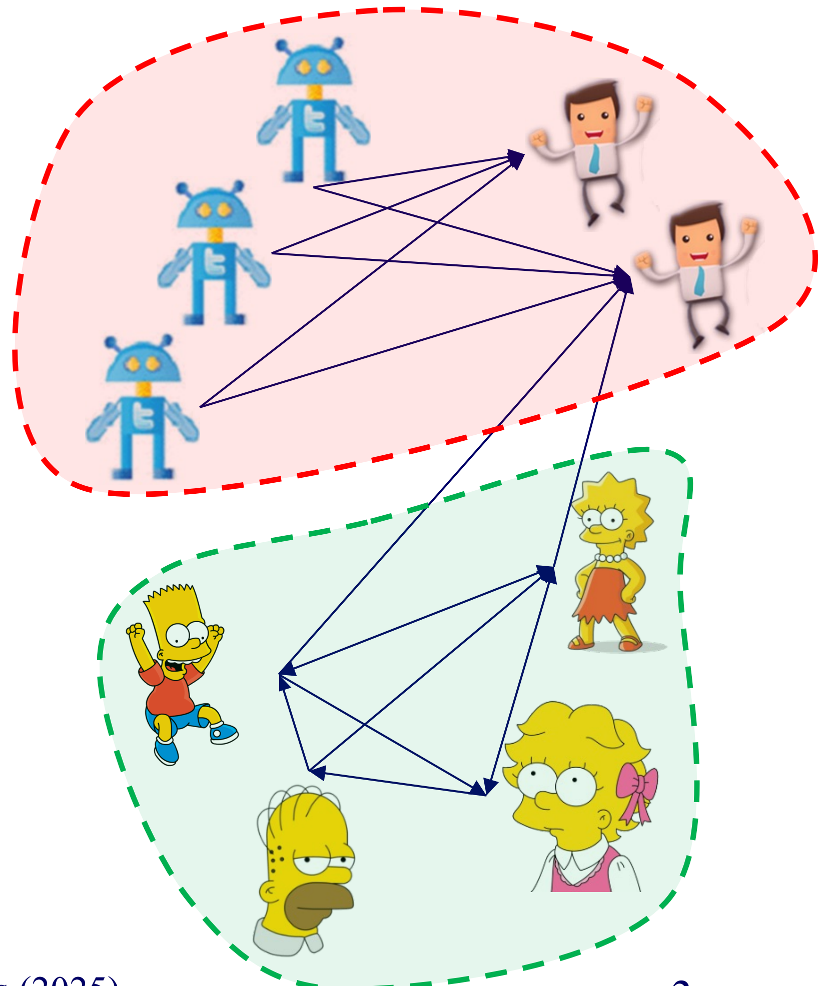
- Introduction – Motivation
- Unsupervised
  - static graphs
  - Time-evolving graphs
- Supervised: BP
- Conclusions

Carnegie Mellon

Fraud Detection: Graph Analysis Problem

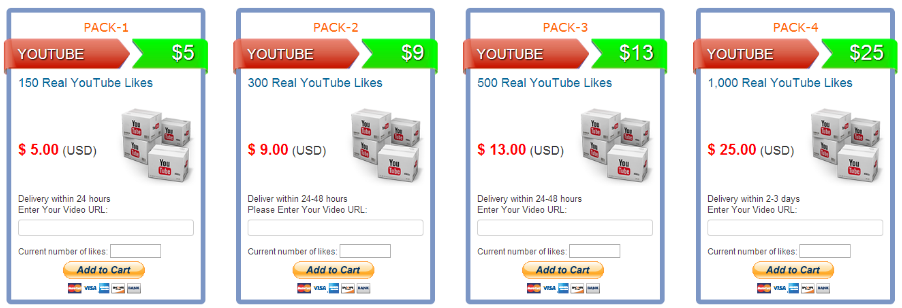
[www.buyfollowz.org]

[buymorelikes.com]

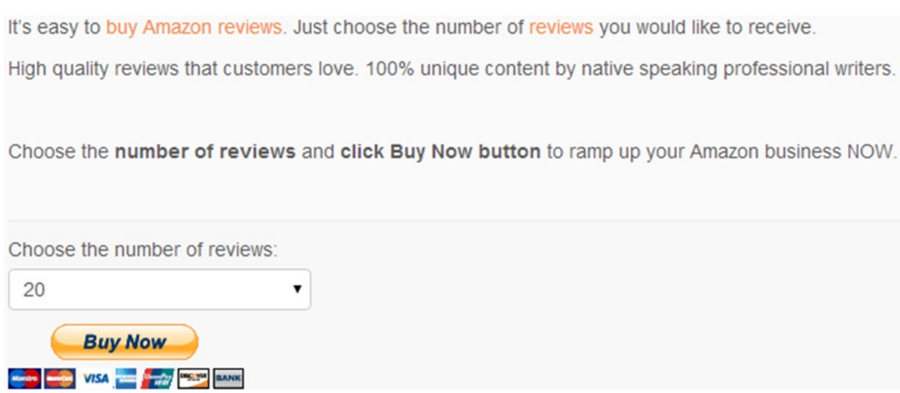


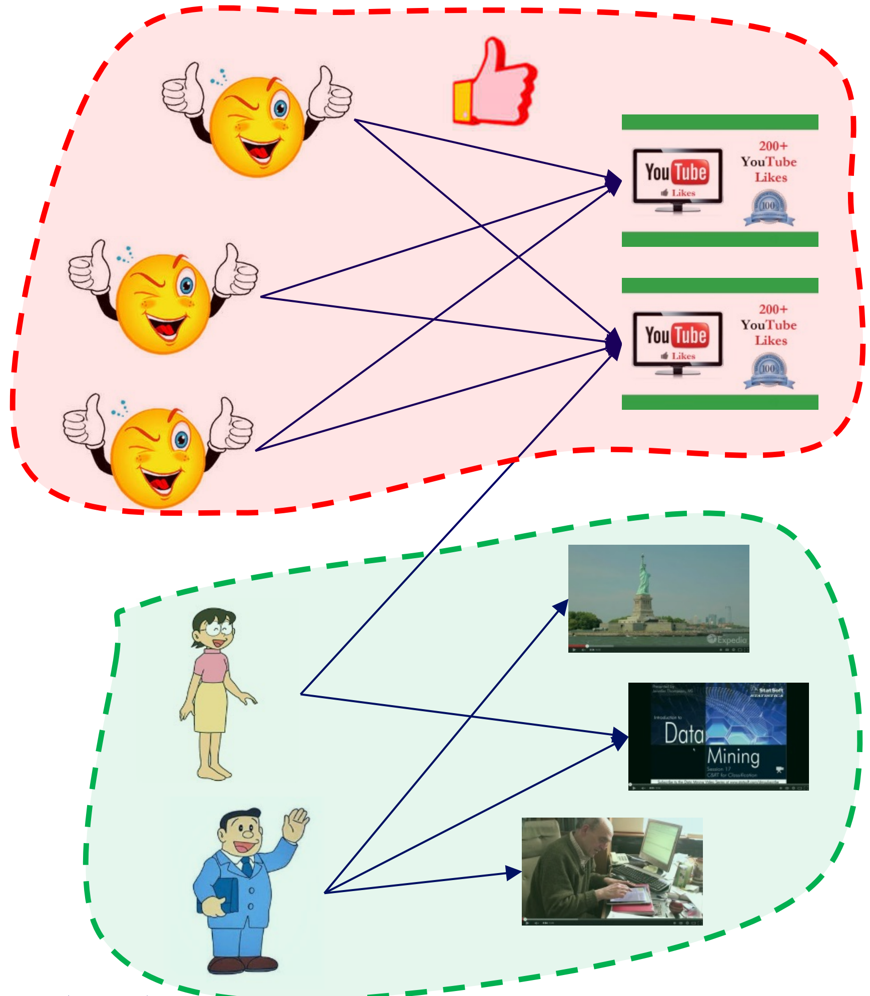
Fraud Detection: Graph Analysis Problem

[buycheaplikes.com]



[reviewsteria.com]





Other applications:

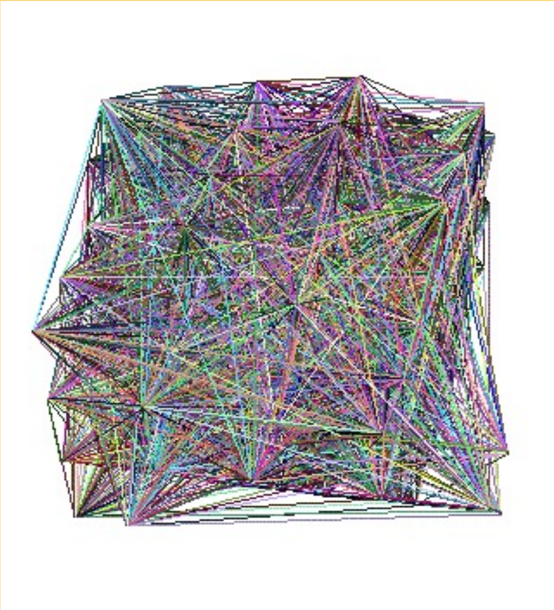
- Cyber-security (DDoS attacks – distributed denial of service)
- Abnormal individuals / social groups
  - dysfunctional departments in a company
  - Onset of depression/hijacking wrt a FB user
  - …

Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
    - Eigenspokes, and extensions
    - Spot strange, individual nodes: oddBall
  - Time-evolving graphs
- Supervised: BP
- Conclusions

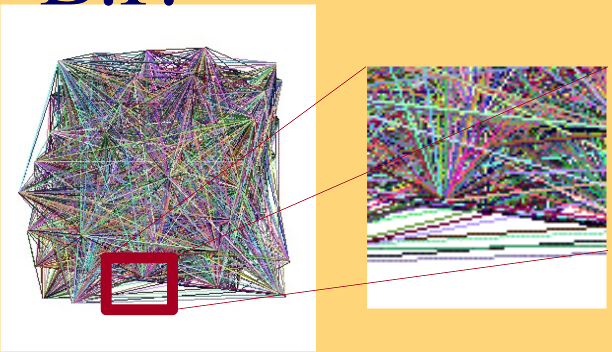
Problem

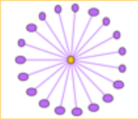
Given:

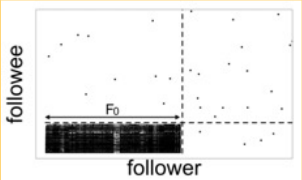


Find:

1) Outliers
2) Lock-step
3) B.P.

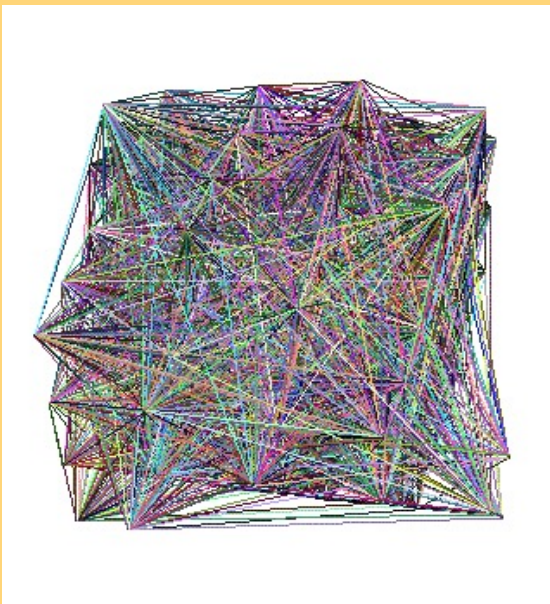






Solutions

Given:

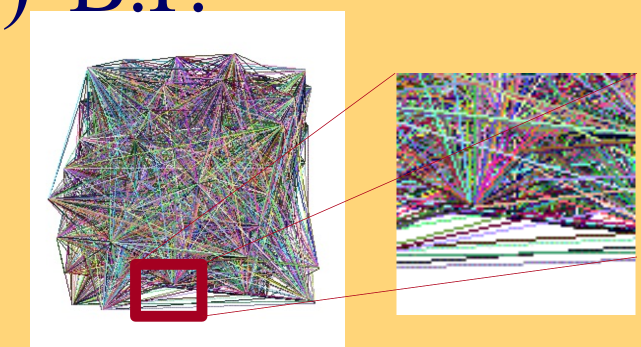


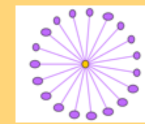
Find:

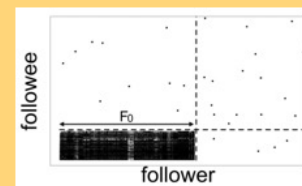
1) Outliers
2) Lock-step
3) B.P.

OddBall

SVD



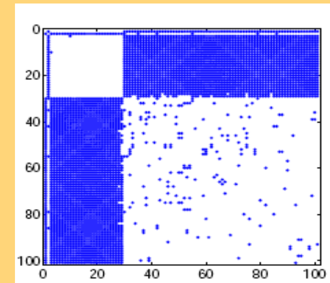




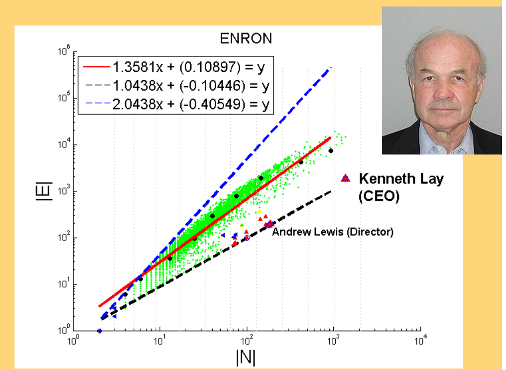
Solutions, cont'd

un-supervised

- Spectral/tensor methods: spot lockstep behavior

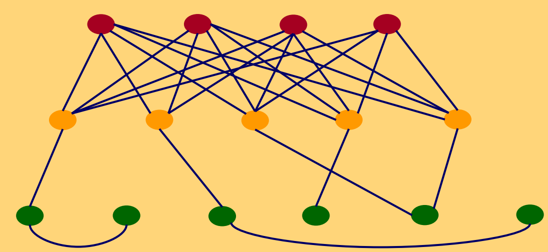


- OddBall: strange, single nodes



semi-supervised

- BP: good for the supervised setting (ie., some labels are given)



EigenSpokes

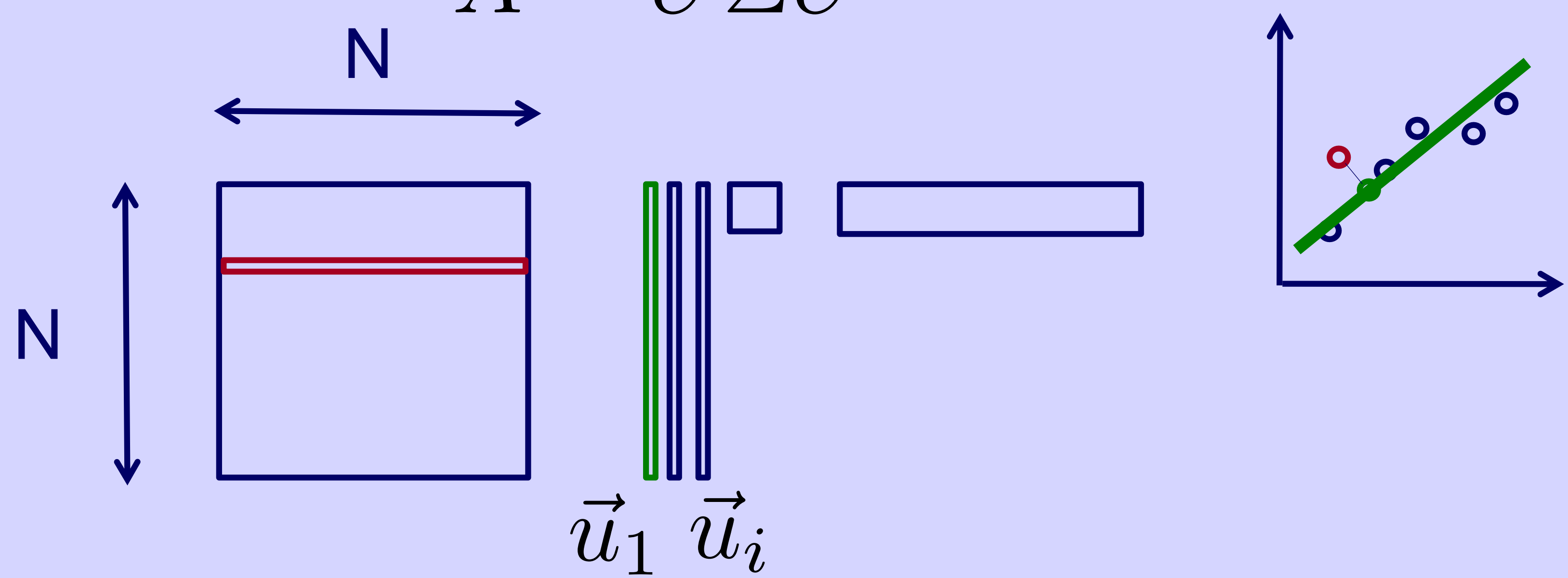
B. Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju and Christos Faloutsos: *EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs,* PAKDD 2010, Hyderabad, India, 21-24 June 2010.

details

EigenSpokes

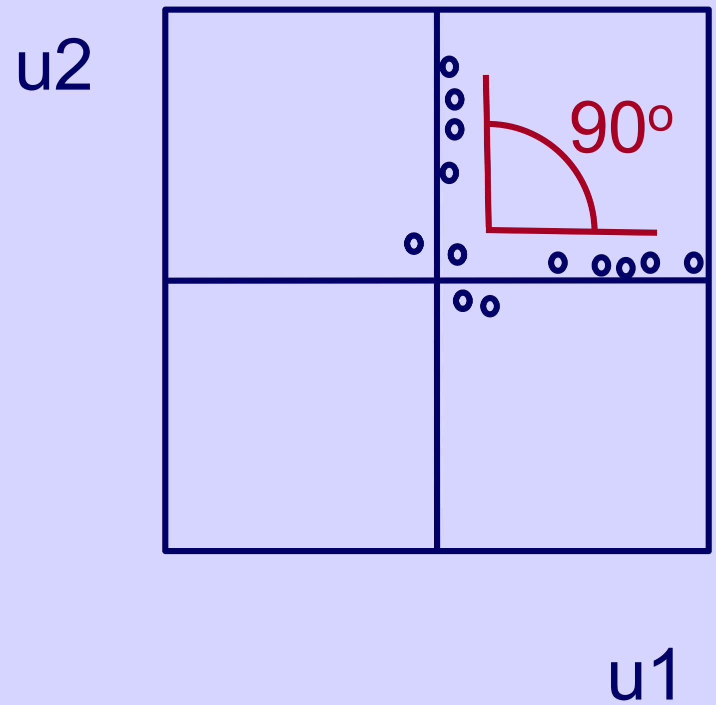
- Eigenvectors of adjacency matrix
  - equivalent to singular vectors (symmetric, undirected graph)

$$A = U \Sigma U^T$$



EigenSpokes

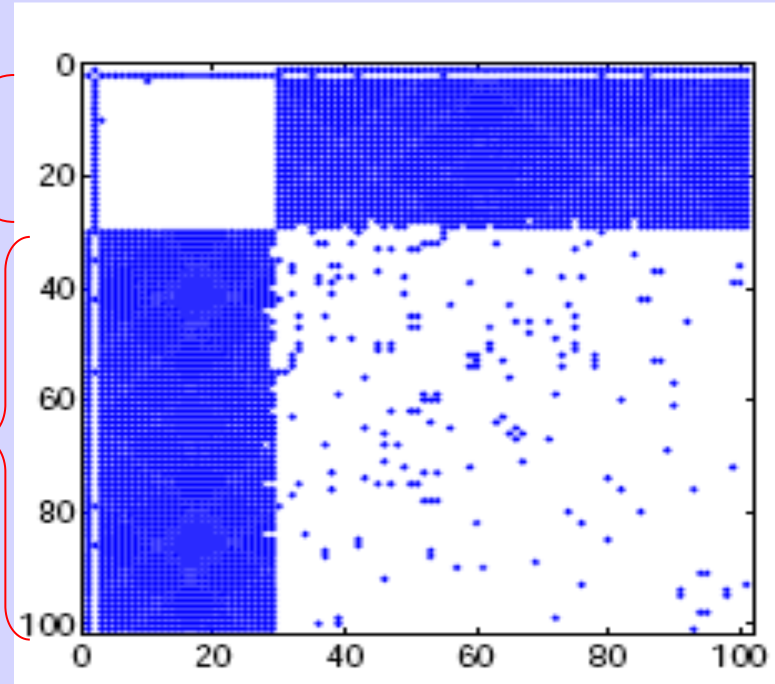
- EE plot:
- Scatter plot of scores of u1 vs u2
- One would expect
  - Many points @ origin
  - A fe[illegible]ttered ~r[illegible]mly



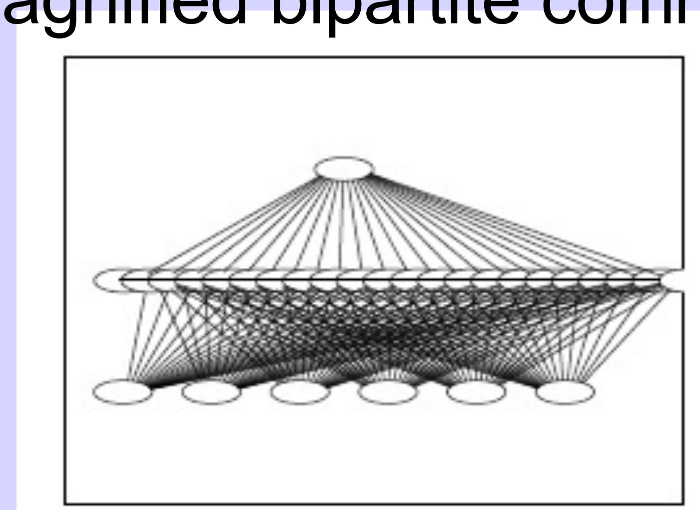
Bipartite Communities!

patents from
same inventor(s)

`cut-and-paste’
bibliography!



magnified bipartite community



Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
    - Eigenspokes, and extensions
    - Spot strange, individual nodes: oddBall
  - Time-evolving graphs
- Supervised: BP
- Conclusions

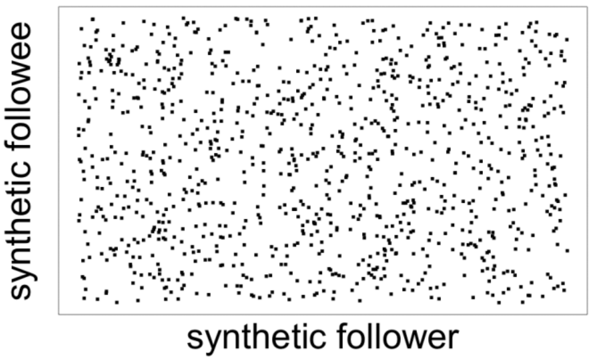
Inferring Strange Behavior from Connectivity Pattern in Social Networks

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

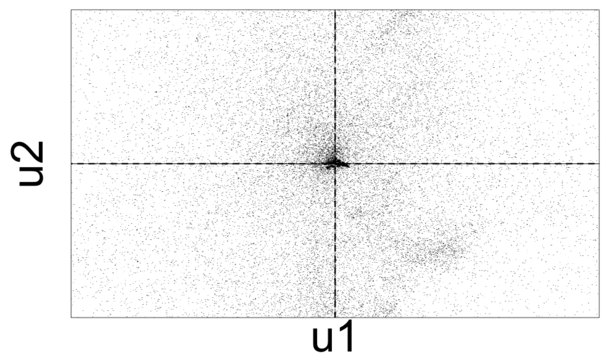
Lockstep and *Spectral Subspace Plot*

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

Adjacency Matrix

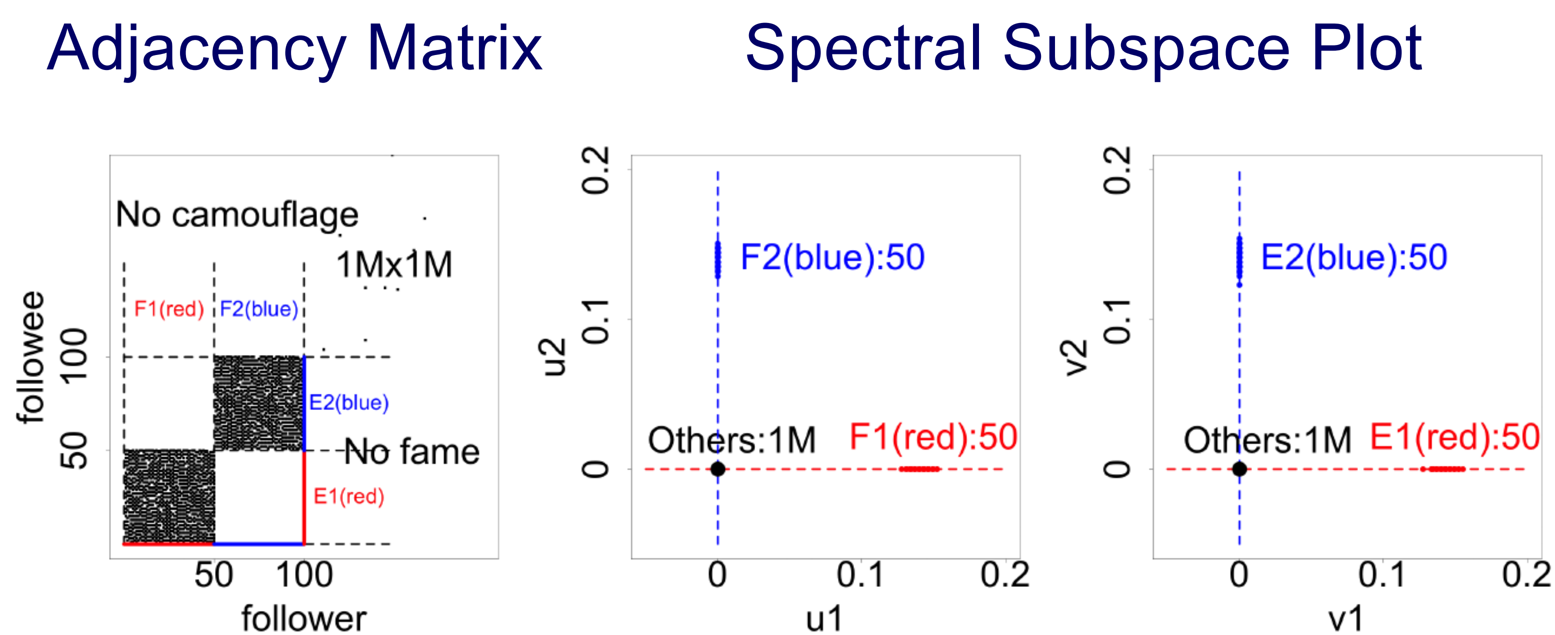


Spectral Subspace Plot



Lockstep and *Spectral Subspace Plot*

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



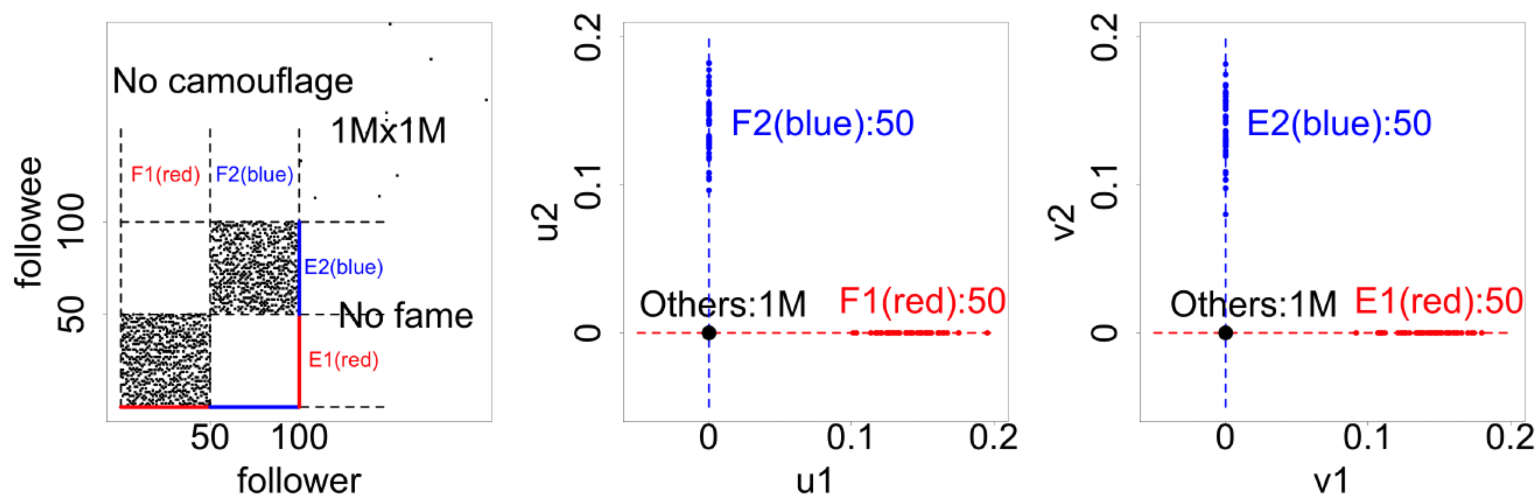
Rule 1 (short "rays"): two blocks, high density (90%), no "camouflage", no "fame"

Carnegie Mellon

Lockstep and *Spectral Subspace Plot*

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

Adjacency Matrix Spectral Subspace Plot



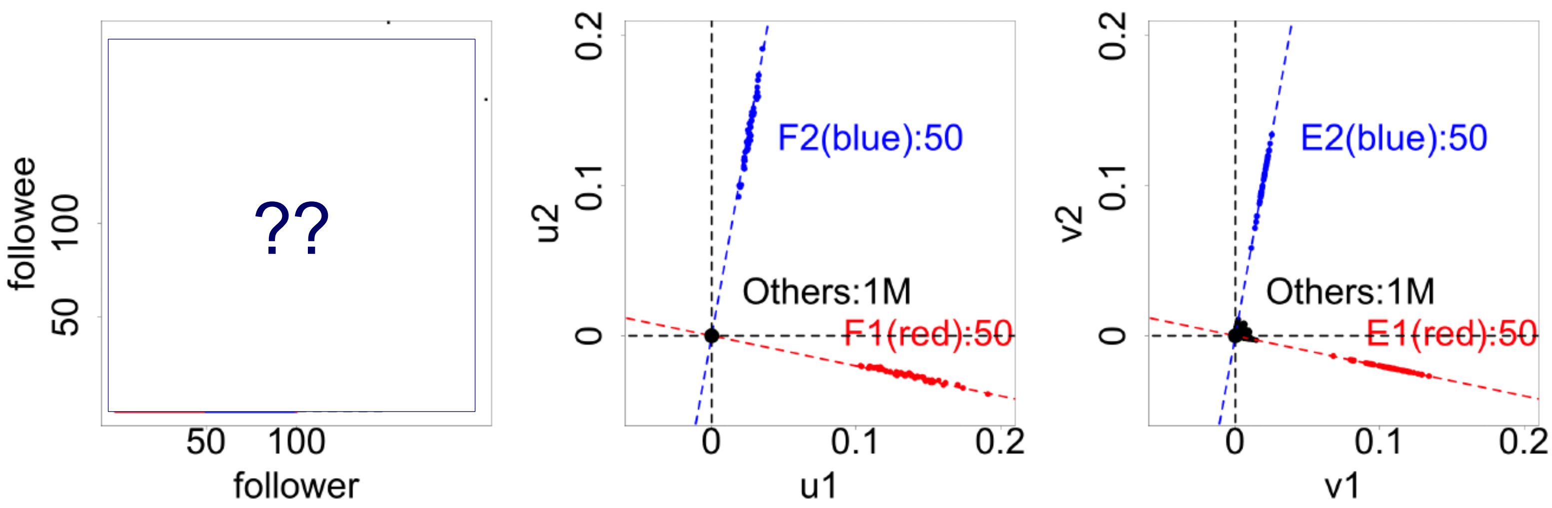
Rule 2 (long "rays"): two blocks, low density (50%), no "camouflage", no "fame"

Lockstep and *Spectral Subspace Plot*

- Case #3: non-overlapping lockstep

Adjacency Matrix

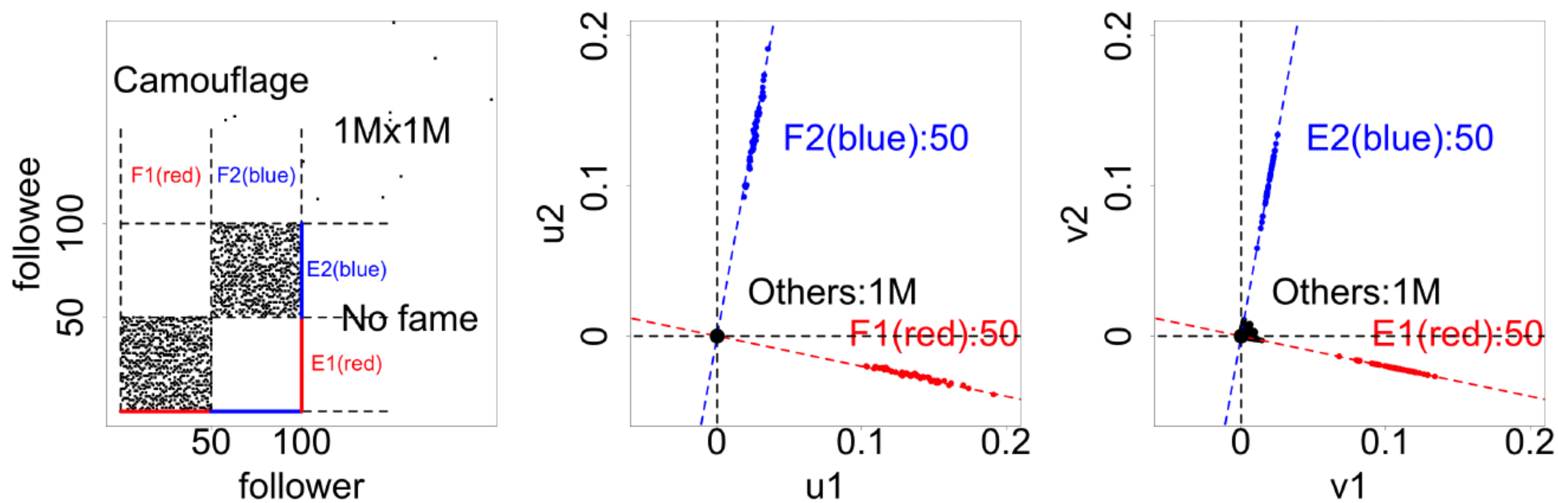
Spectral Subspace Plot



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"

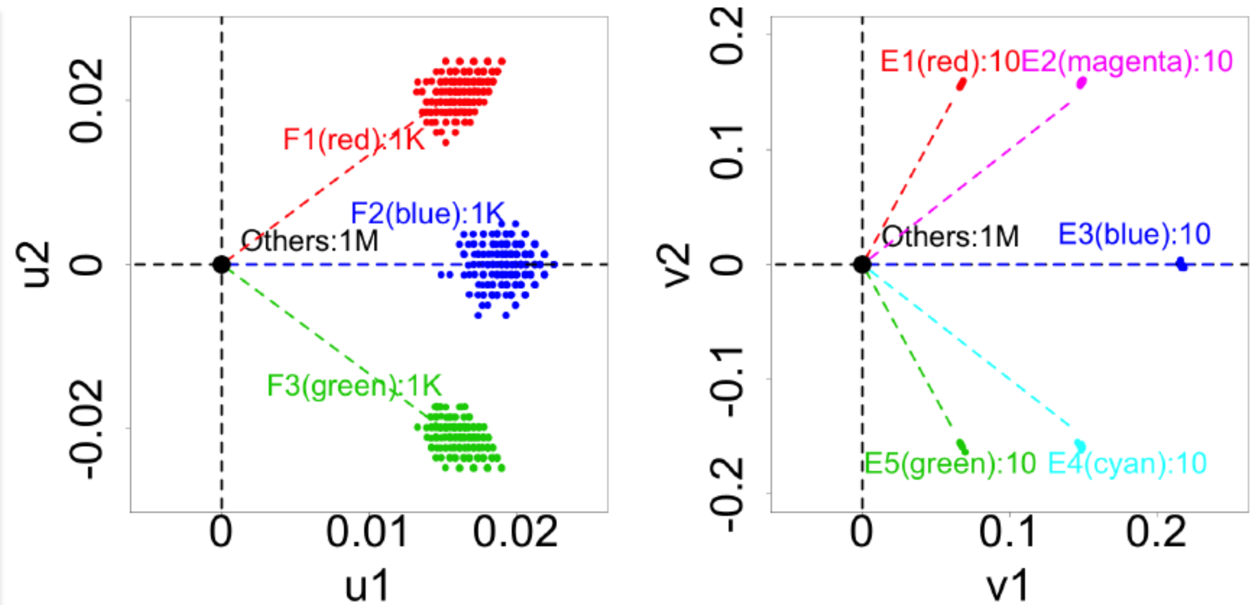
Lockstep and *Spectral Subspace Plot*

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

Adjacency Matrix

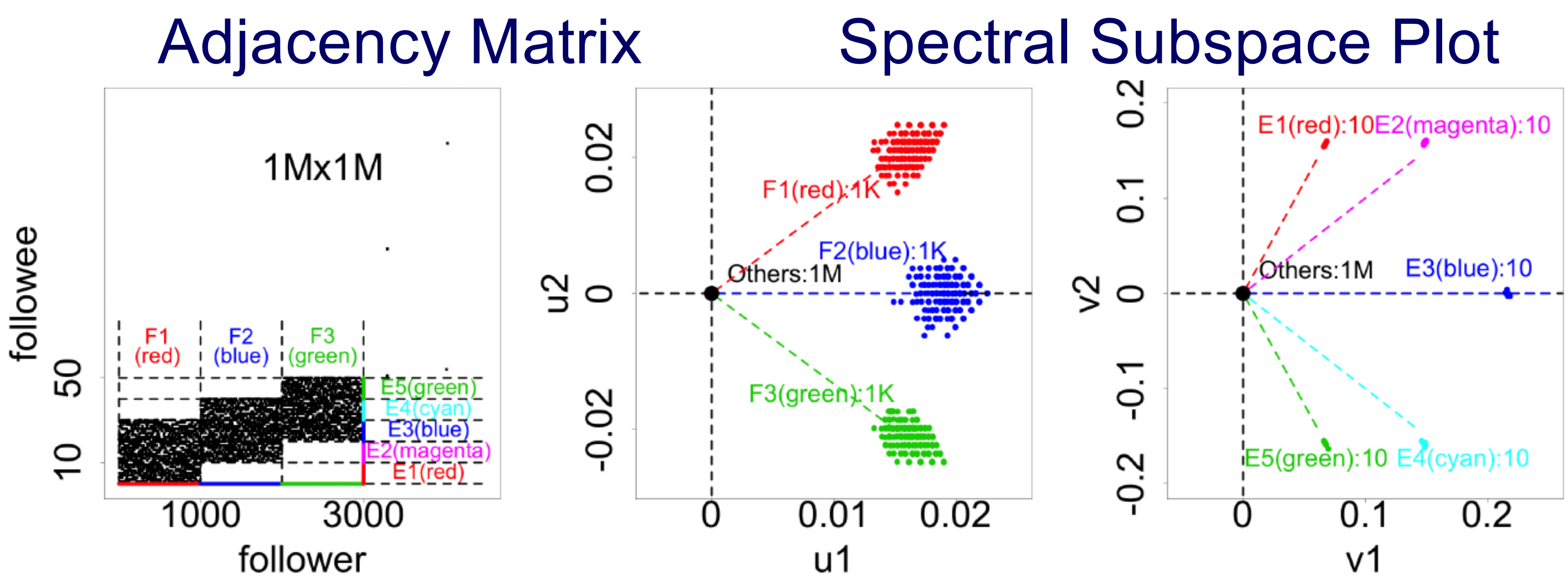
?

Spectral Subspace Plot



Lockstep and *Spectral Subspace Plot*

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



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

Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
    - Eigenspokes, and extensions
    - Spot strange, individual nodes: oddBall
  - Time-evolving graphs
- Supervised: BP
- Conclusions

Carnegie Mellon

OddBall: Spotting Anomalies in Weighted Graphs

Leman Akoglu, Mary McGlohon, Christos Faloutsos

Carnegie Mellon University

School of Computer Science

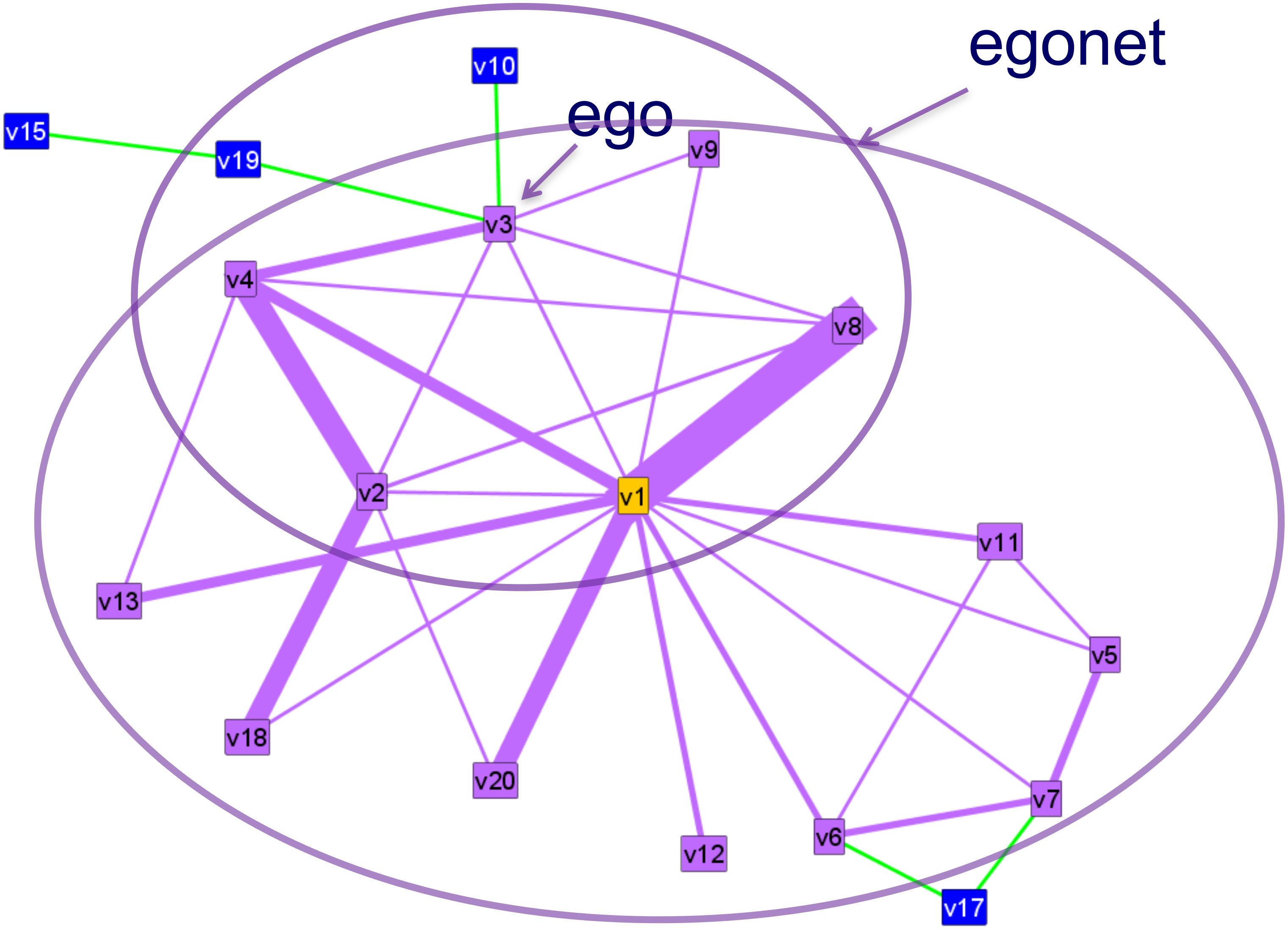
PAKDD 2010, Hyderabad, India

Main idea

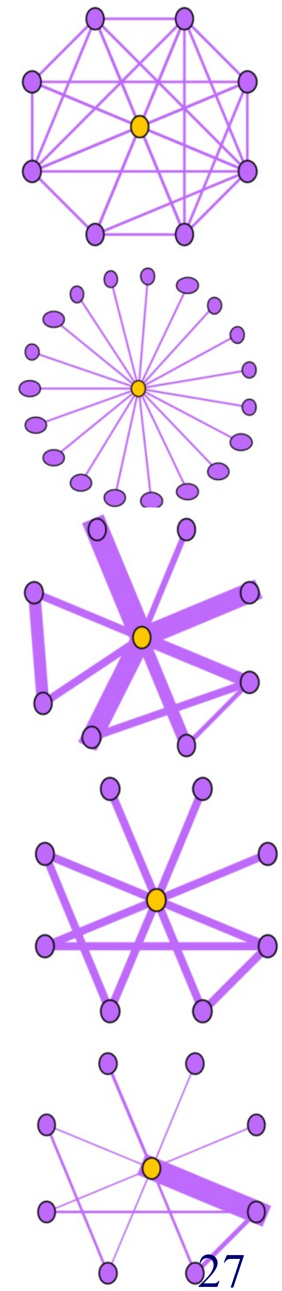
For each node,

- extract 'ego-net' (=1-step-away neighbors)
- Extract features (#edges, total weight, etc etc)
- Compare with the rest of the population

What is an egonet?

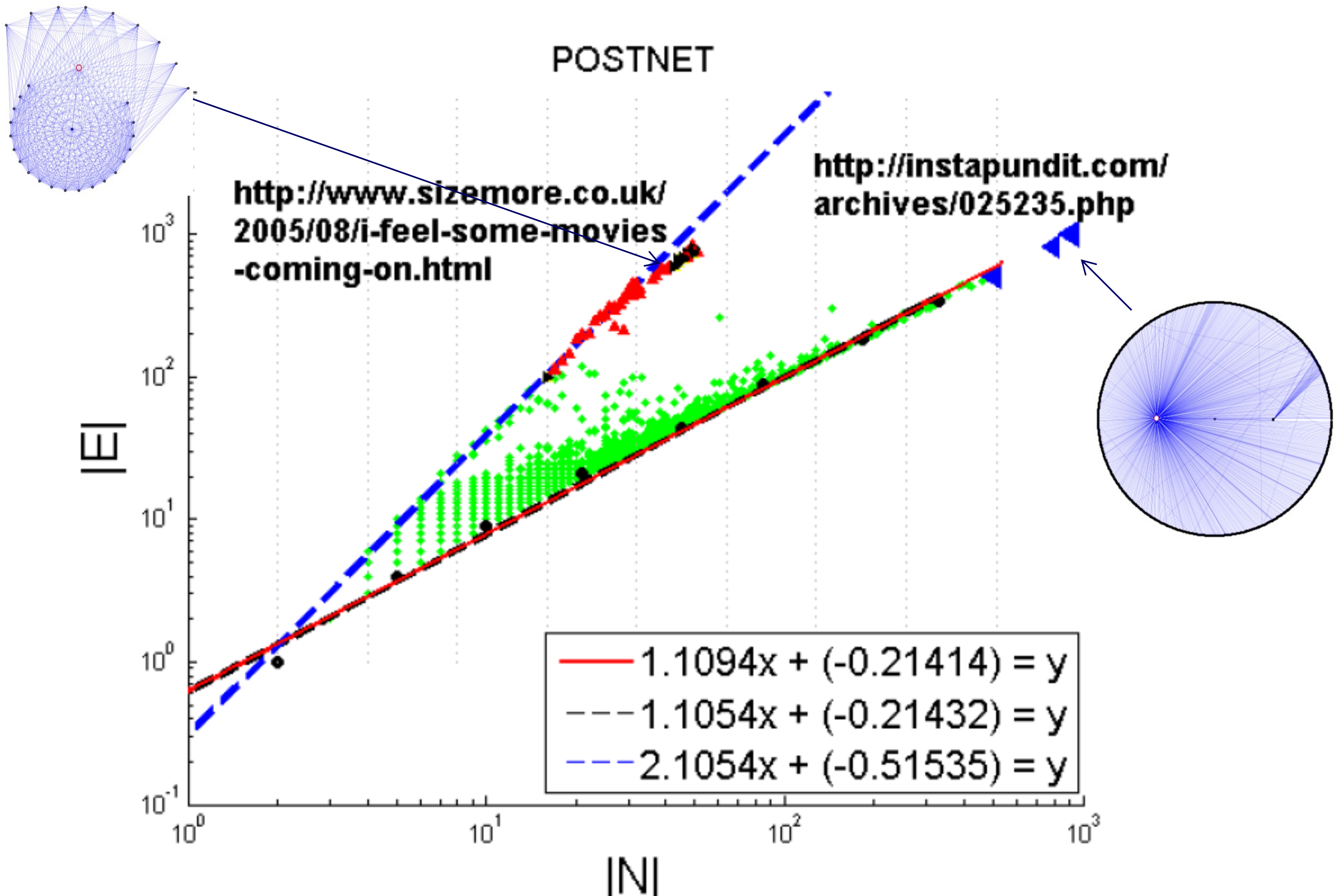


Selected Features

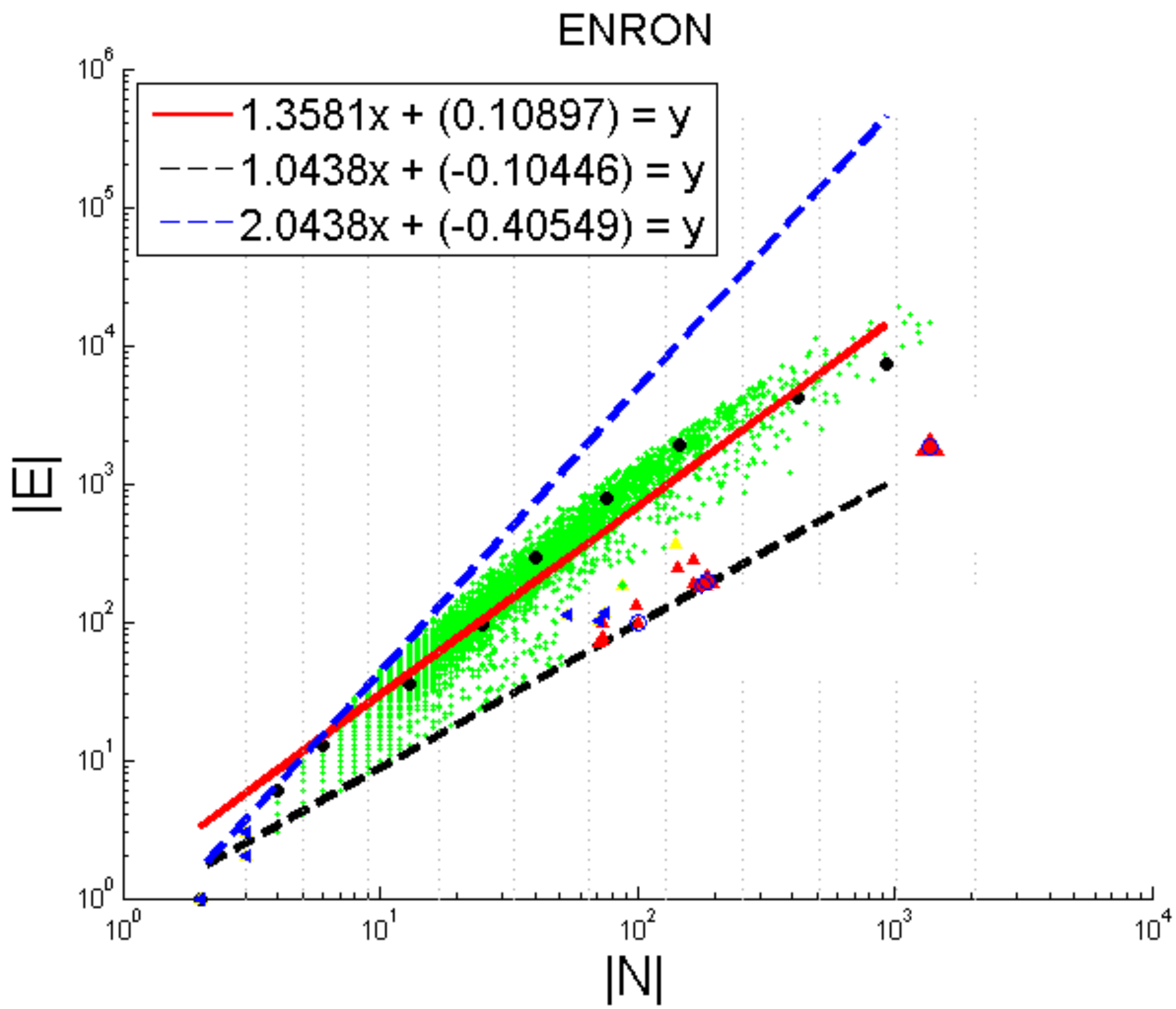


- N_i: number of neighbors (degree) of ego i
- E_i: number of edges in egonet i
- W_i: total weight of egonet i
- $\lambda_{w,i}$: principal eigenvalue of the weighted adjacency matrix of egonet I

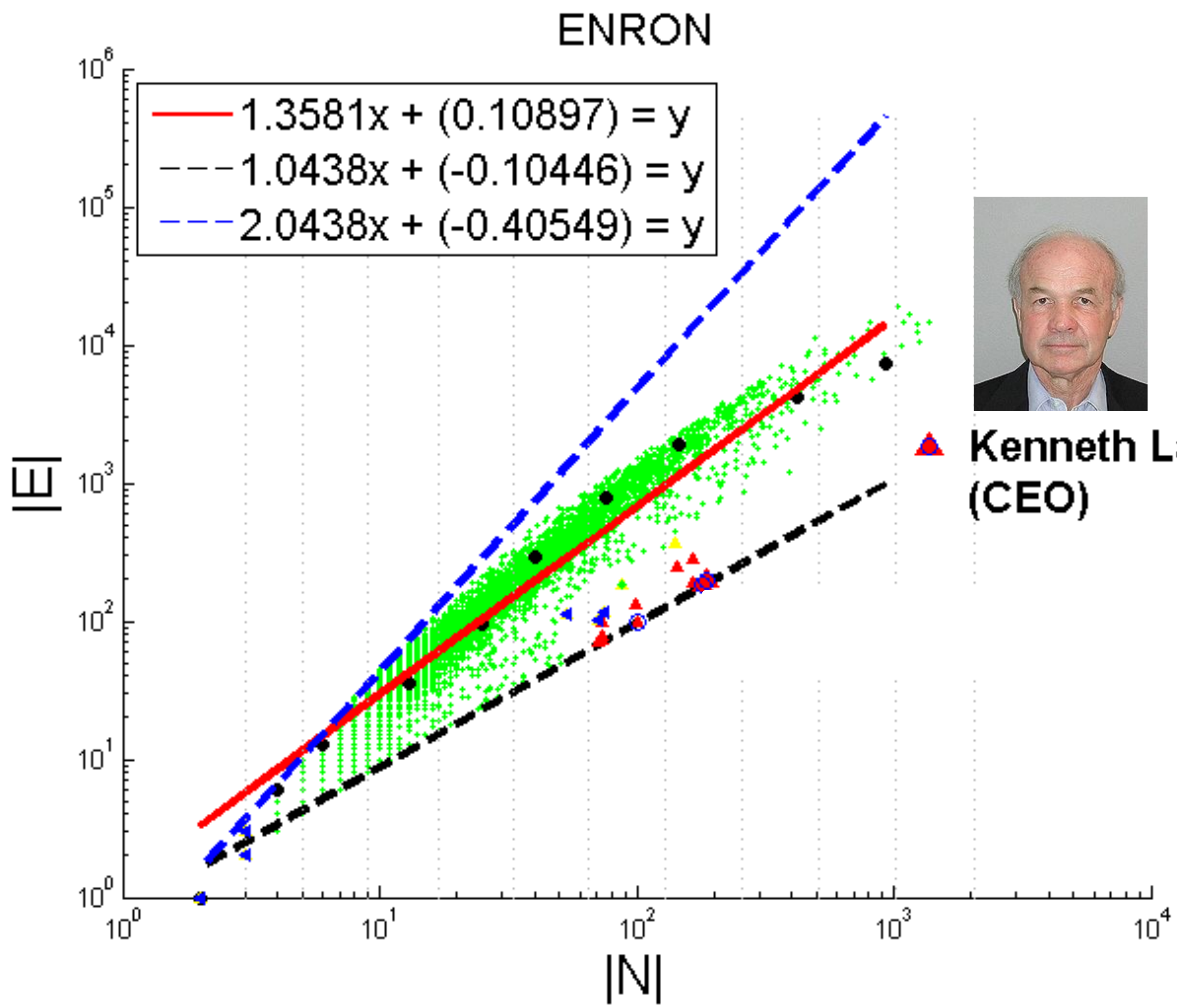
Near-Clique/Star



Near-Clique/Star

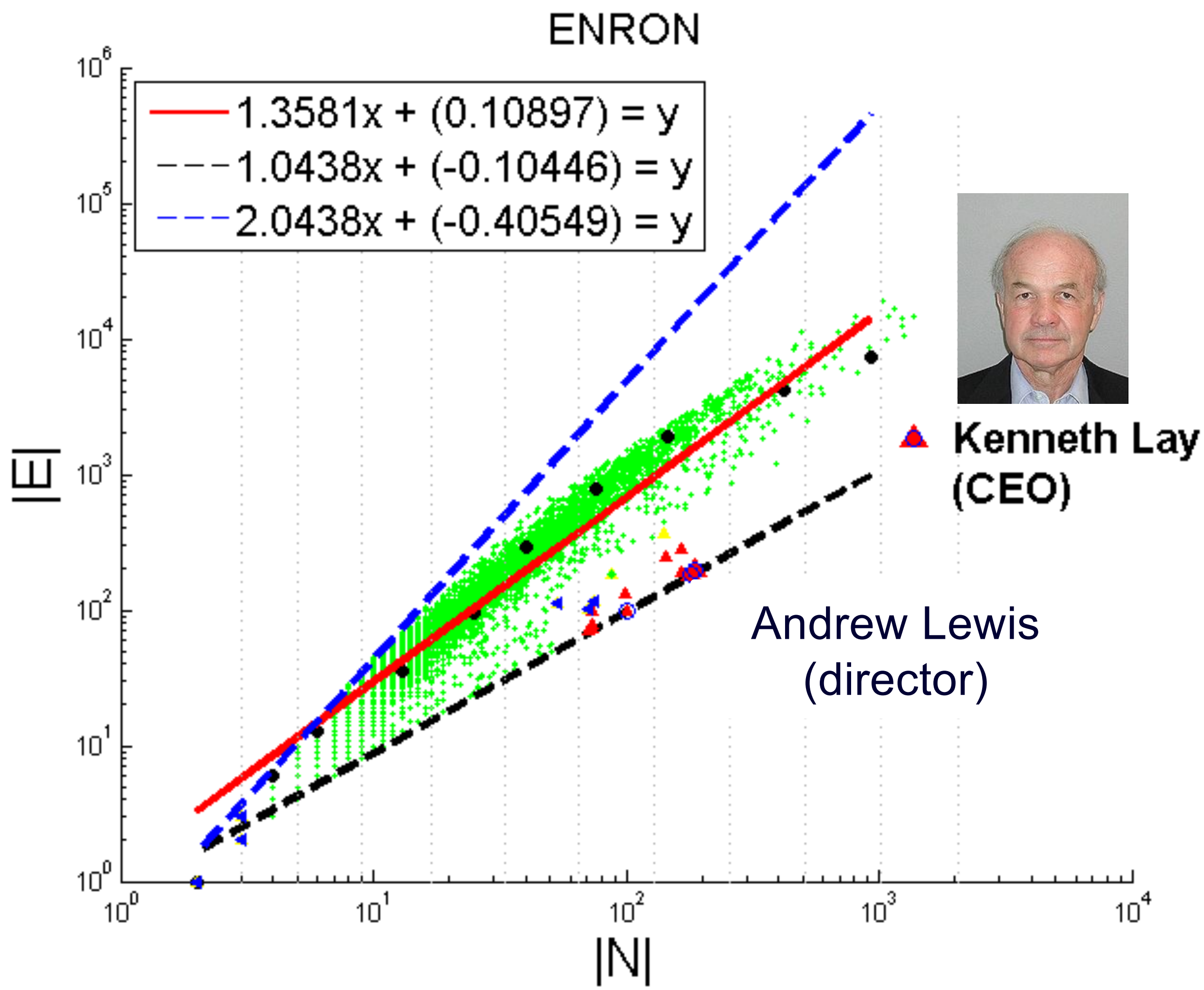


Near-Clique/Star



Kenneth Lay (CEO)

Near-Clique/Star



Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
  - Time-evolving graphs
    - ‘copyCatch’ in FaceBook
    - Patterns of IAT
    - [tensors]
- Supervised: BP
- Conclusions

Carnegie Mellon

Fraud

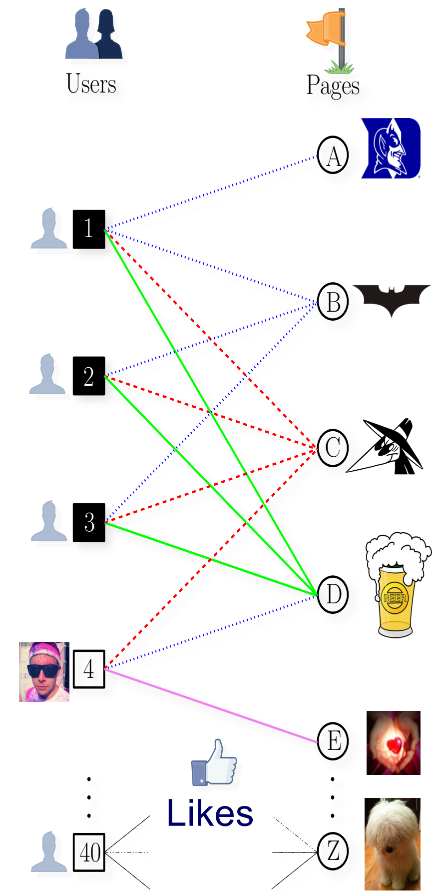
- Given
  - Who ‘likes’ what page, and when
- Find
  - Suspicious users and suspicious products

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

Carnegie Mellon

Fraud

- Given
  - Who ‘likes’ what page, and when
- Find
  - Suspicious users and suspicious products



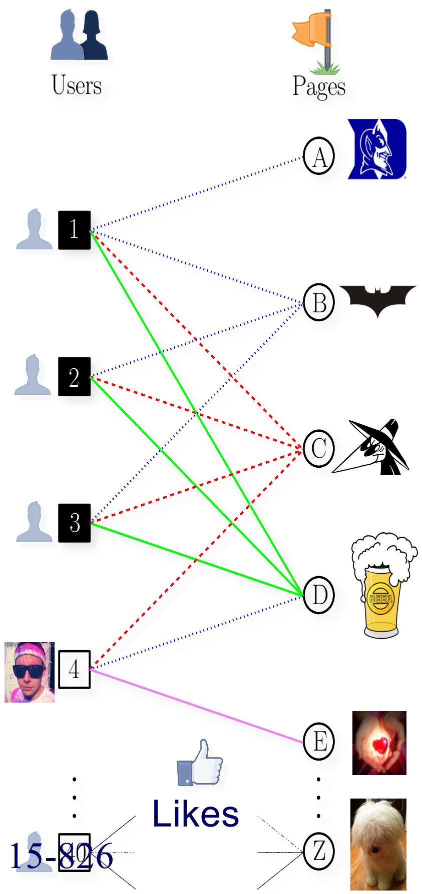
CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks, Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, Christos Faloutsos *WWW, 2013.*

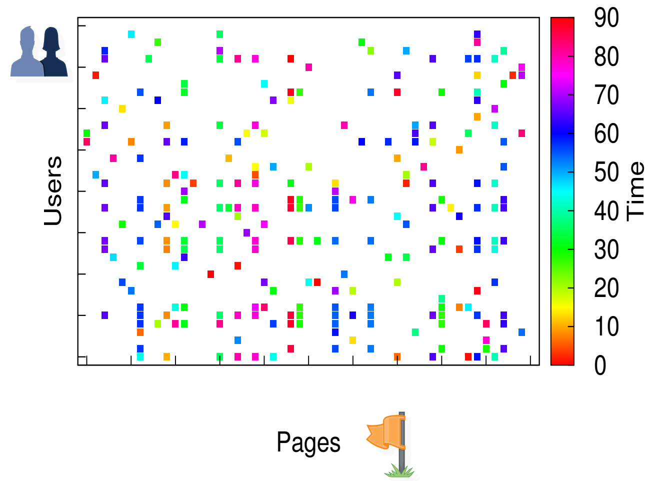
Carnegie Mellon

Graph Patterns and Lockstep Behavior

Our intuition

Lockstep behavior: Same Likes, same time

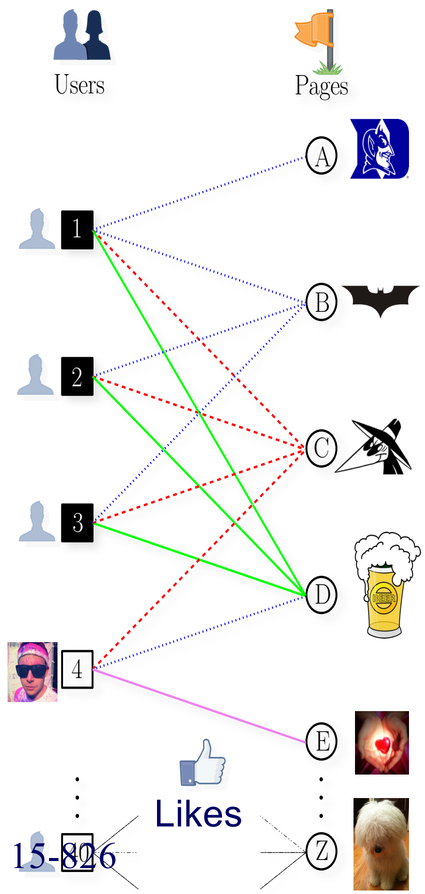


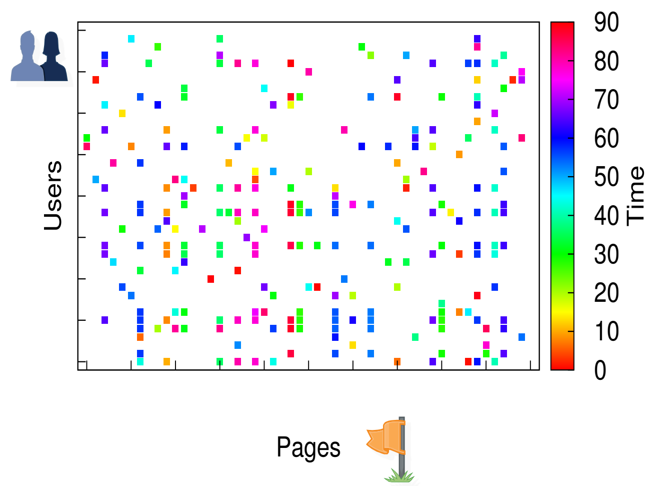


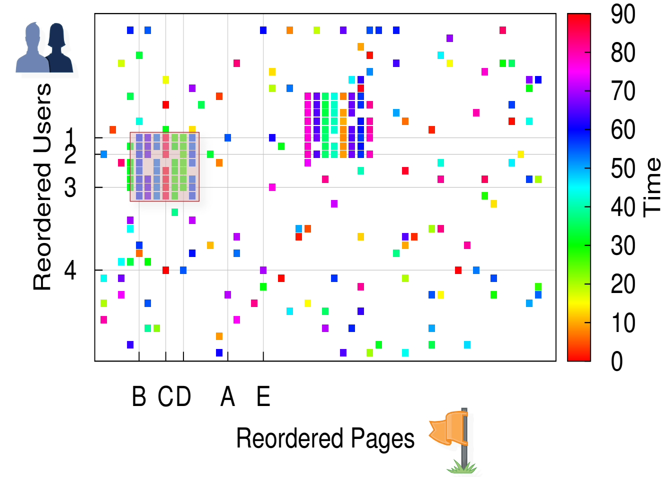
Graph Patterns and Lockstep Behavior

Our intuition

Lockstep behavior: Same Likes, same time



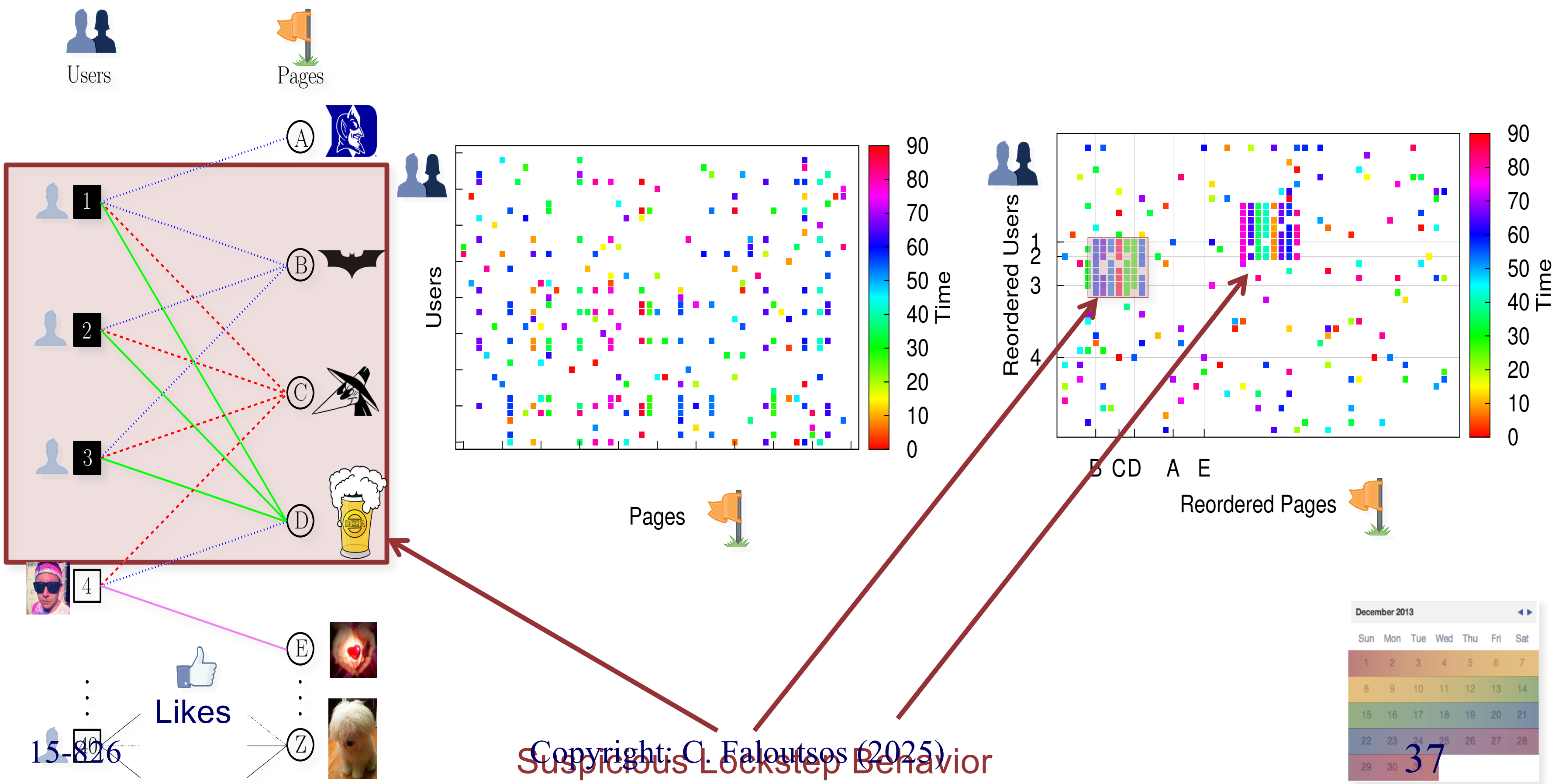




Graph Patterns and Lockstep Behavior

Our intuition

Lockstep behavior: Same Likes, same time

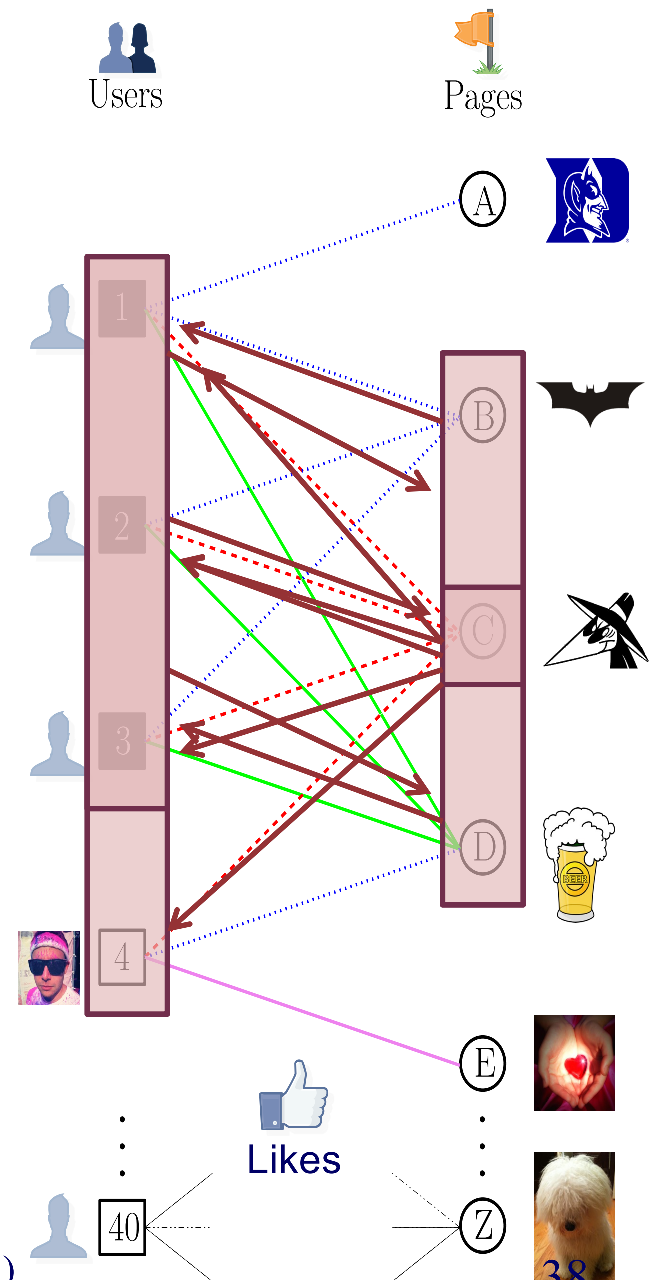


Suspicious Lockstep Behavior

MapReduce Overview

Use Hadoop to search for many clusters in parallel:

1. Start with randomly seed
2. Update set of Pages and center Like times for each cluster
3. Repeat until convergence

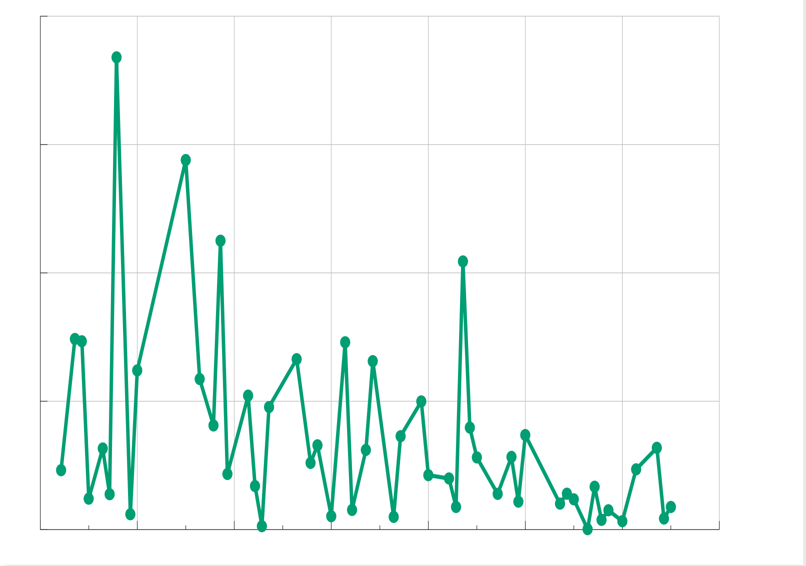


Deployment at Facebook

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

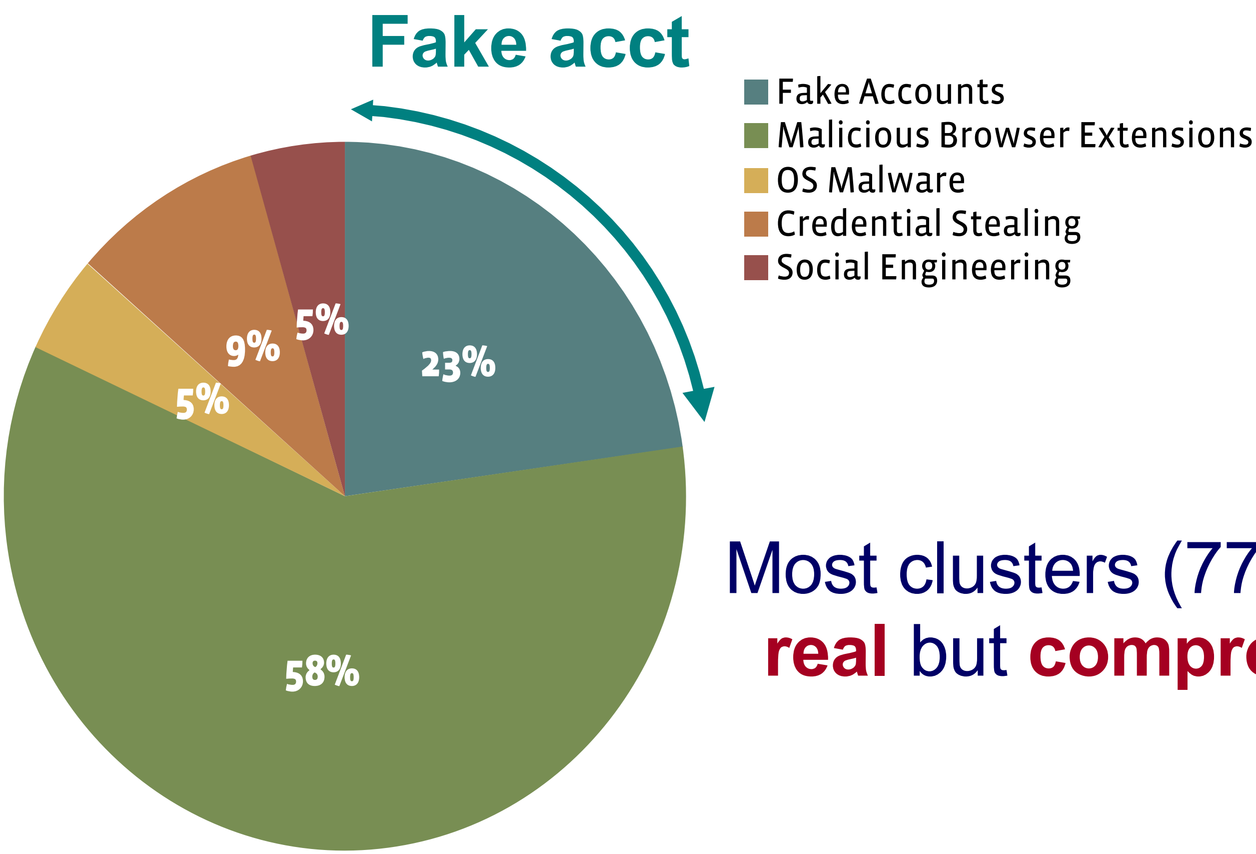
3 months of *CopyCatch* @ Facebook

#users caught



time

Deployment at Facebook



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

Manually labeled 22 randomly selected *clusters* from February 2013

Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
  - Time-evolving graphs
    - ‘copyCatch’ in FaceBook
    - Patterns of IAT
    - [tensors]
- Supervised: BP
- Conclusions

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

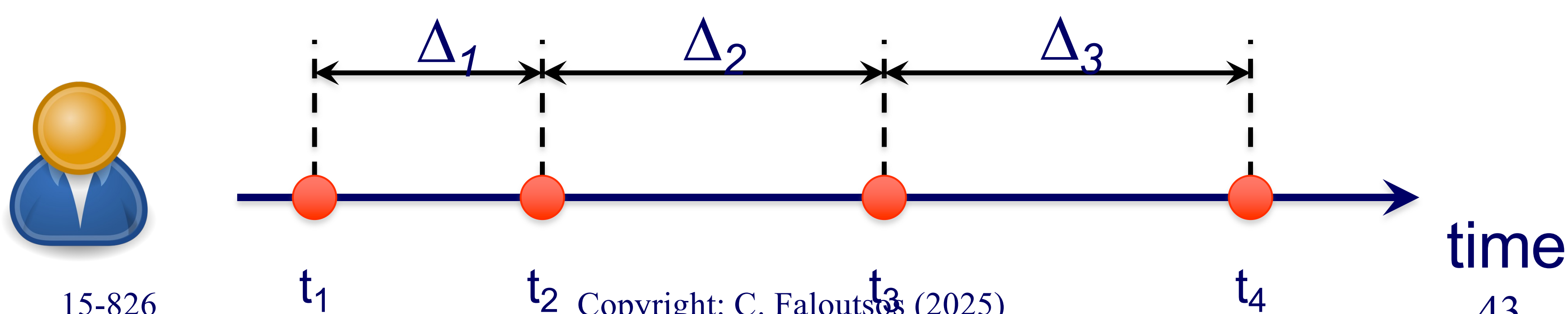
*alceufc@icmc.usp.br

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:

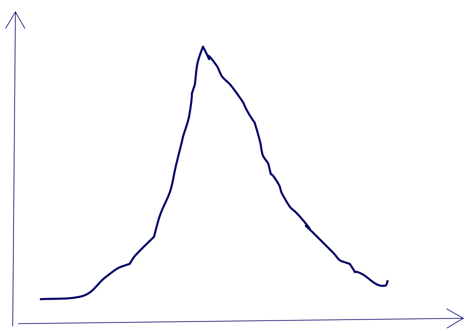
- Sequence of postings time-stamps: $\boldsymbol{T} = (t_1, t_2, t_3, \ldots)$
- Inter-arrival times (IAT) of postings: $(\Delta_1, \Delta_2, \Delta_3, \ldots)$

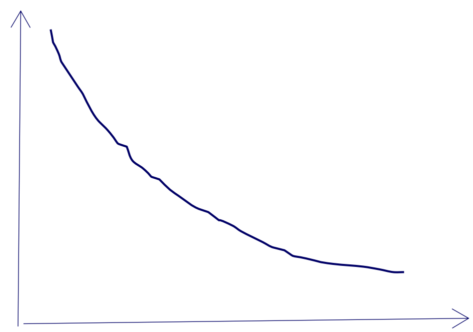


Guess the IAT pdf

- Gaussian (10’ +/- a bit)?
- Power law? Slope?
- Log-logistic?
- Periodic (spike for 1day, 1 week)?

PDF

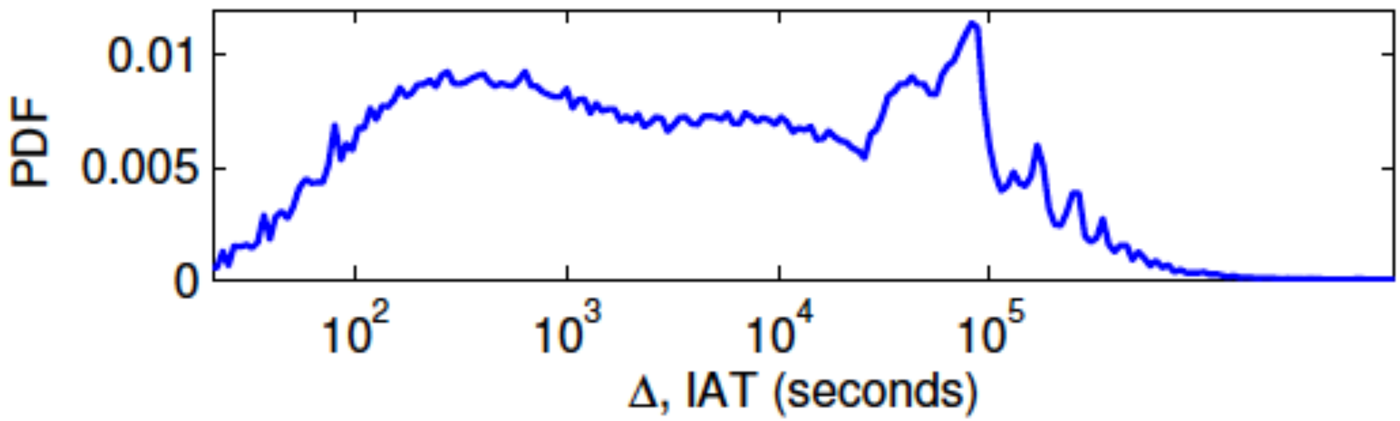




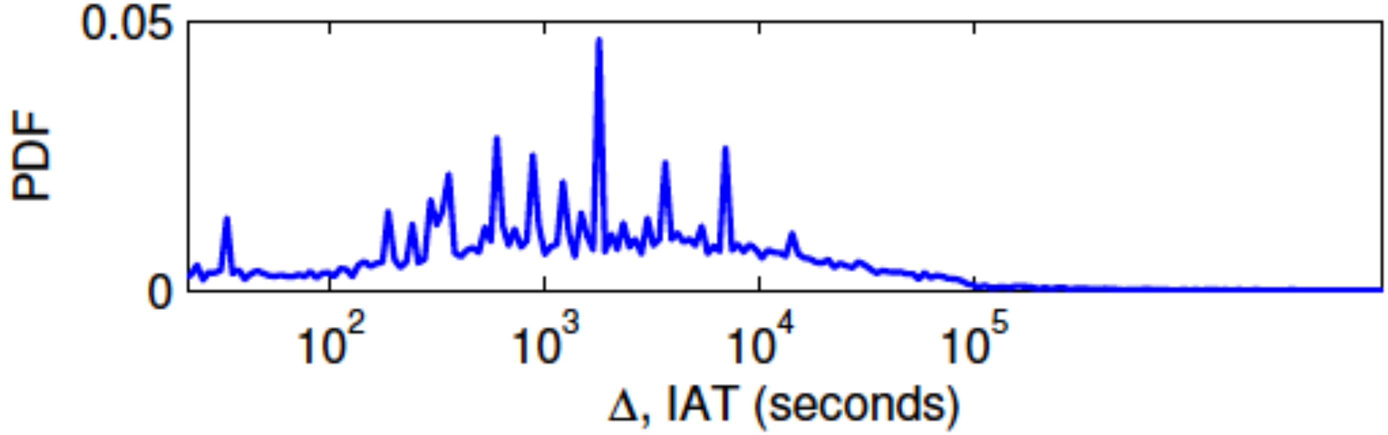
IAT

Human? Robots?

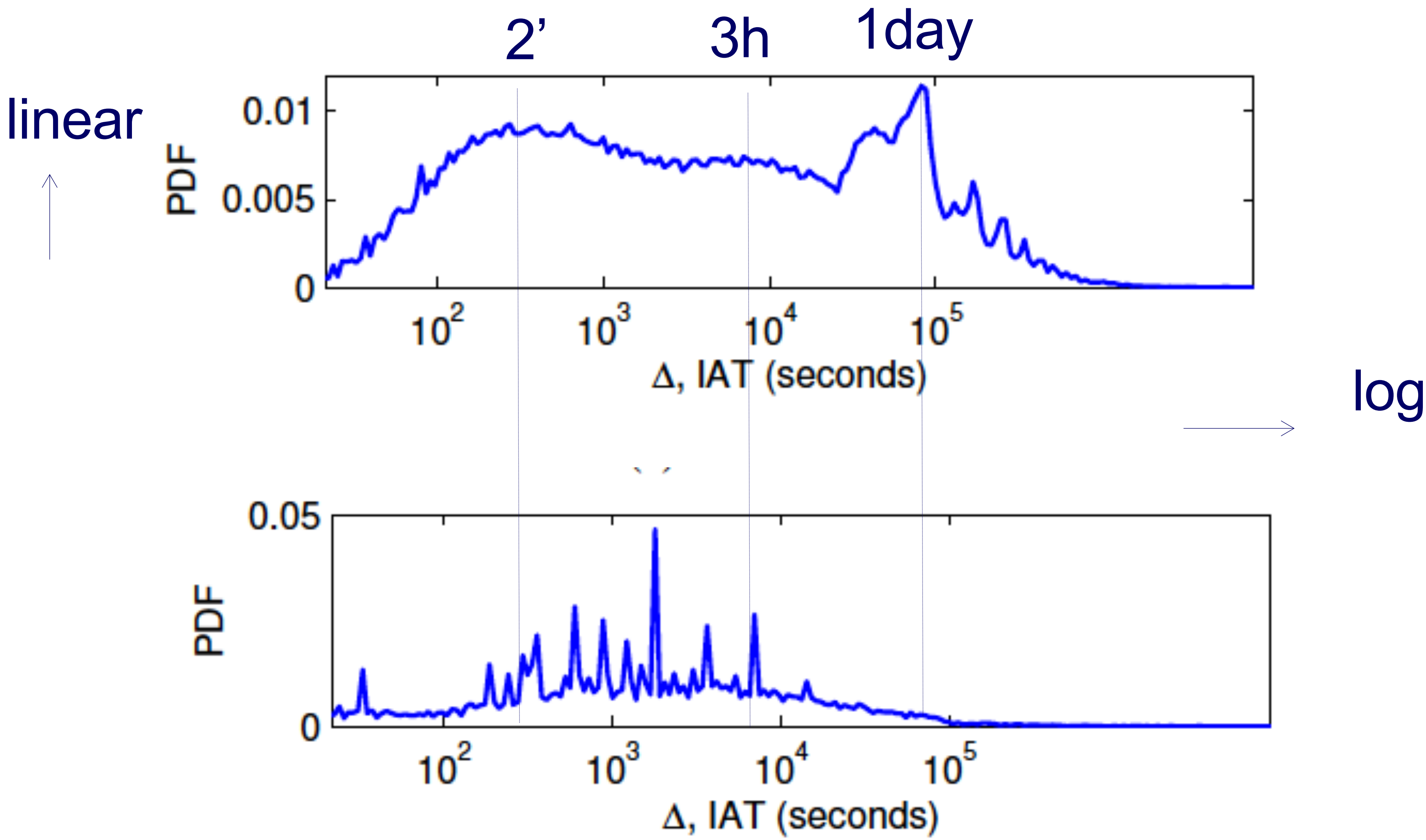
linear



log



Human? Robots?

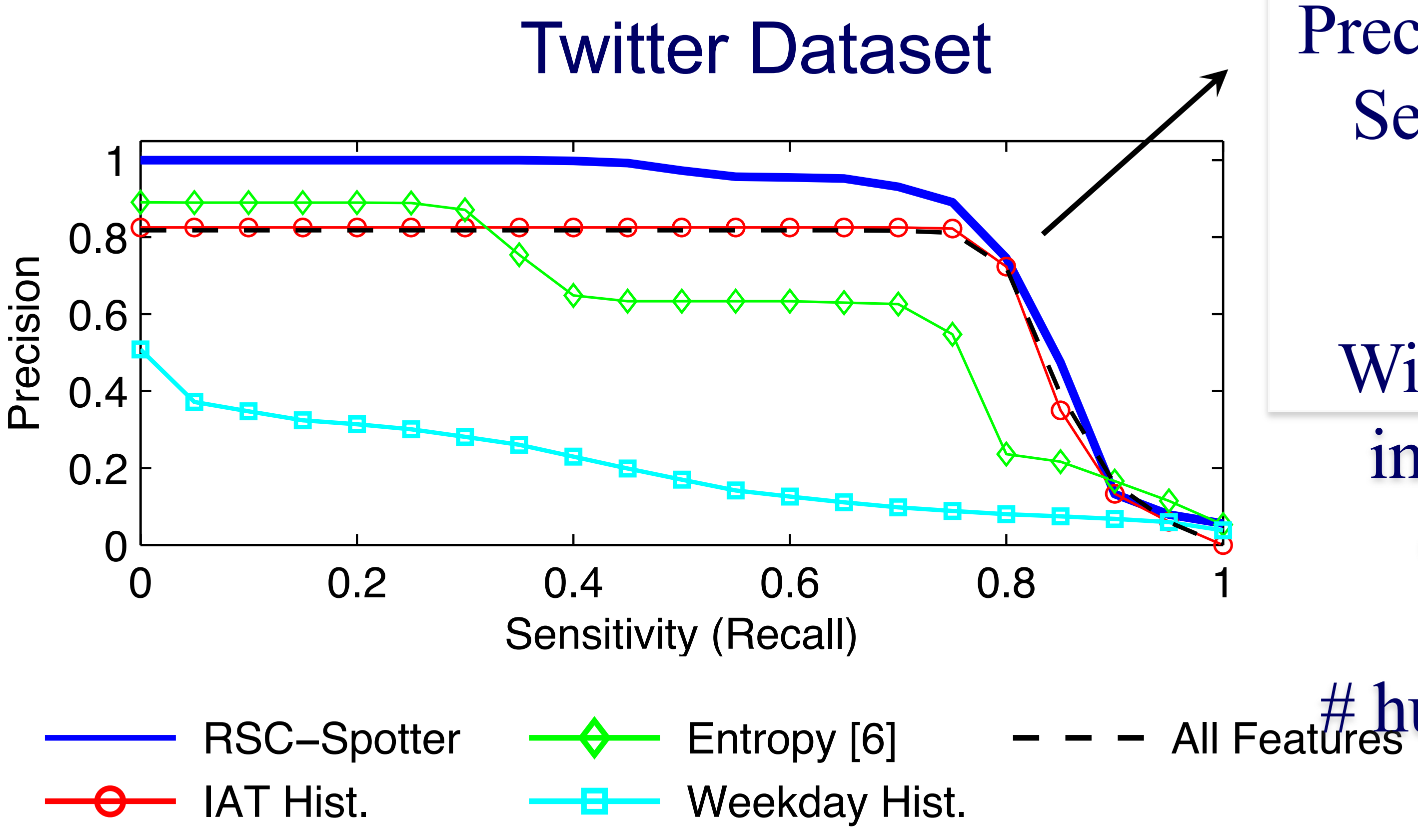


Carnegie Mellon

Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top



Precision > 94%
Sensitivity > 70%

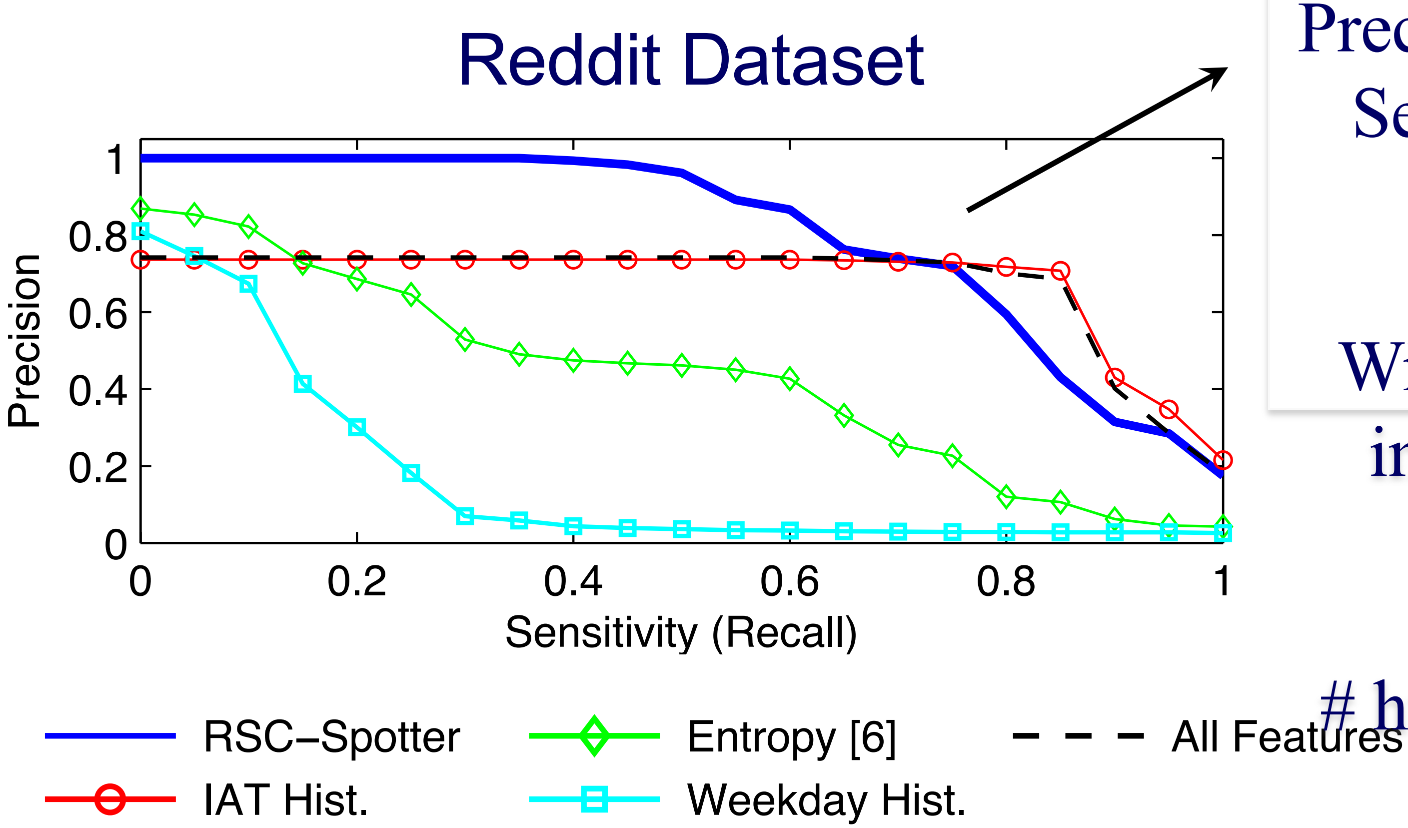
With strongly imbalanced datasets

humans >> # bots

Experiments: Can RSC-Spotter Detect Bots?

Precision vs. Sensitivity Curves

Good performance: curve close to the top



Precision > 96%
Sensitivity > 47%

With strongly imbalanced datasets

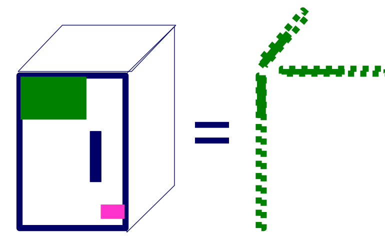
humans >> # bots

Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
  - Time-evolving graphs
    - ‘copyCatch’ in FaceBook
    - Patterns of IAT
    - [tensors]
- Supervised: BP
- Conclusions

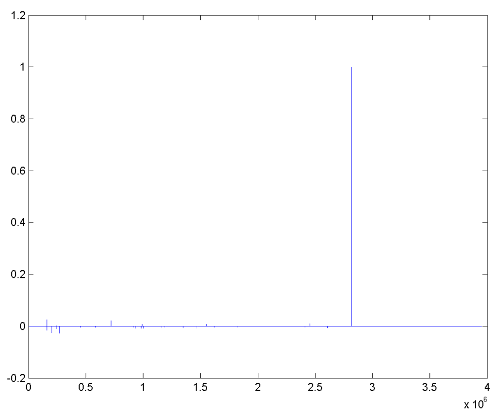
CarnegieMellon

Anomaly detection in time-evolving graphs

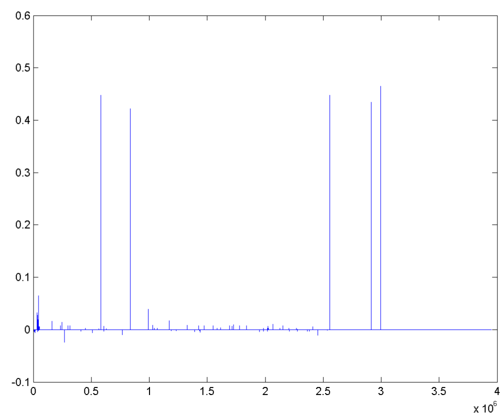


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

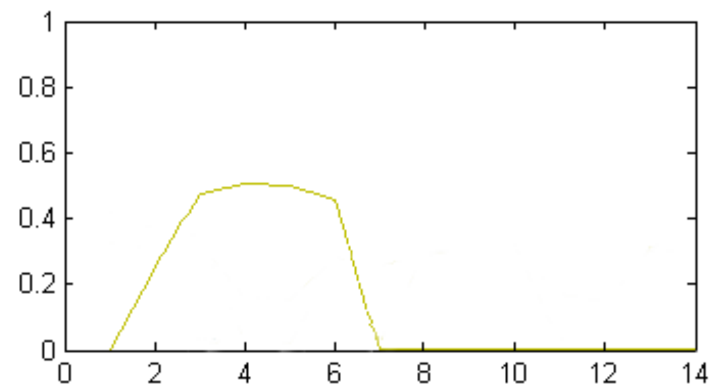
1 caller



5 receivers

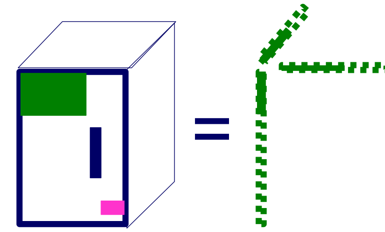


4 days of activity



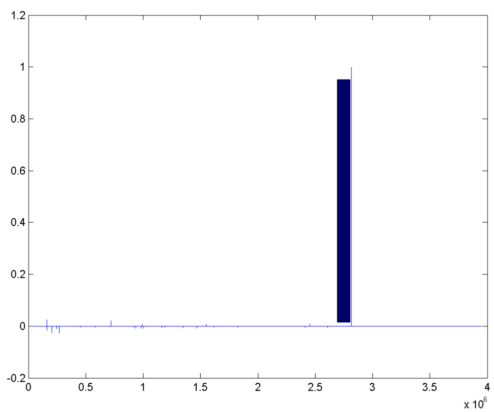
~200 calls to EACH receiver on EACH day!

Anomaly detection in time-evolving graphs

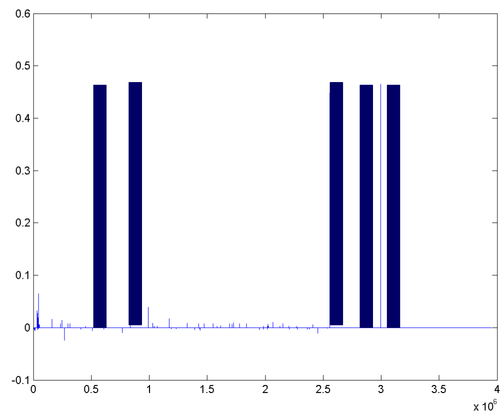


- Anomalous communities in phone call data:
  - European country, 4M clients, data over 2 weeks

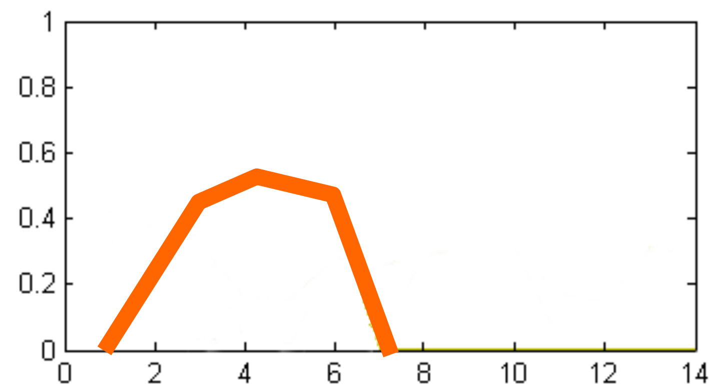
1 caller



5 receivers



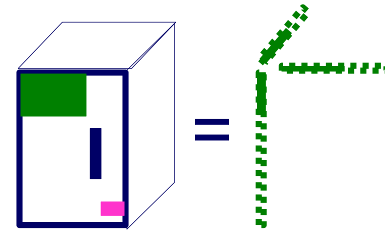
4 days of activity



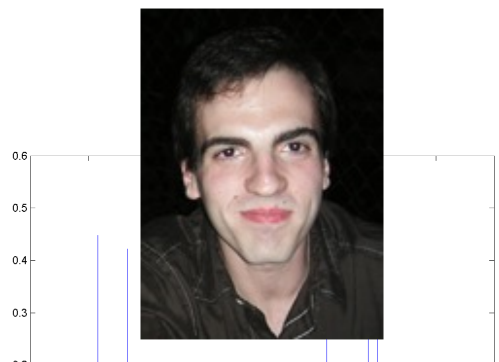
~200 calls to EACH receiver on EACH day!

Carnegie Mellon

Anomaly detection in time-evolving graphs



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

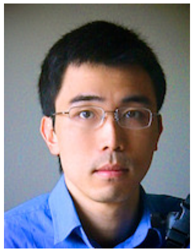


Miguel Araujo, Spiros Papadimitriou, Stephan Günnemann, Christos Faloutsos, Prithwish Basu, Ananthram Swami, Evangelos Papalexakis, Danai Koutra. *Com2: Fast Automatic Discovery of Temporal (Comet) Communities*. PAKDD 2014, Tainan, Taiwan.

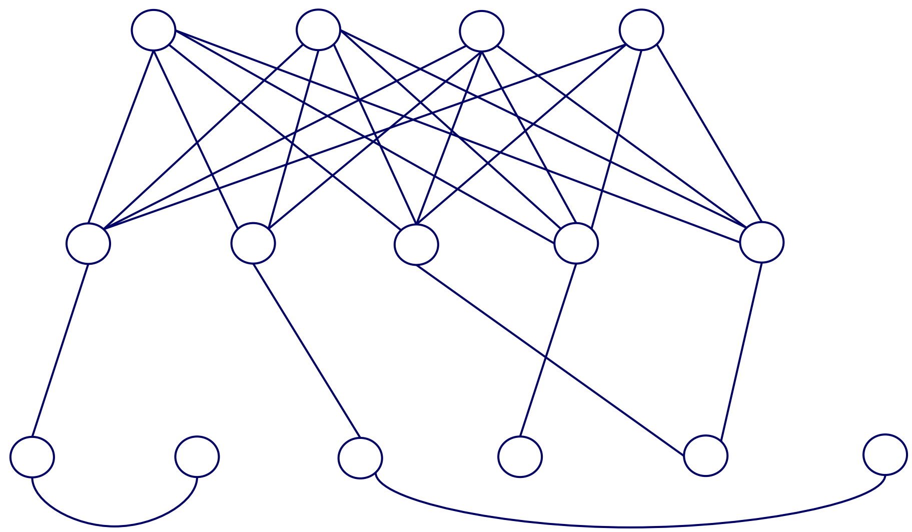
Roadmap

- Introduction – Motivation
- Unsupervised
  - static graphs
  - Time-evolving graphs
- Supervised: BP
  - Ebay fraud ('NetProbe')
  - Malware (Polonium)
- Conclusions

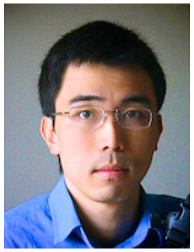
E-bay Fraud detection

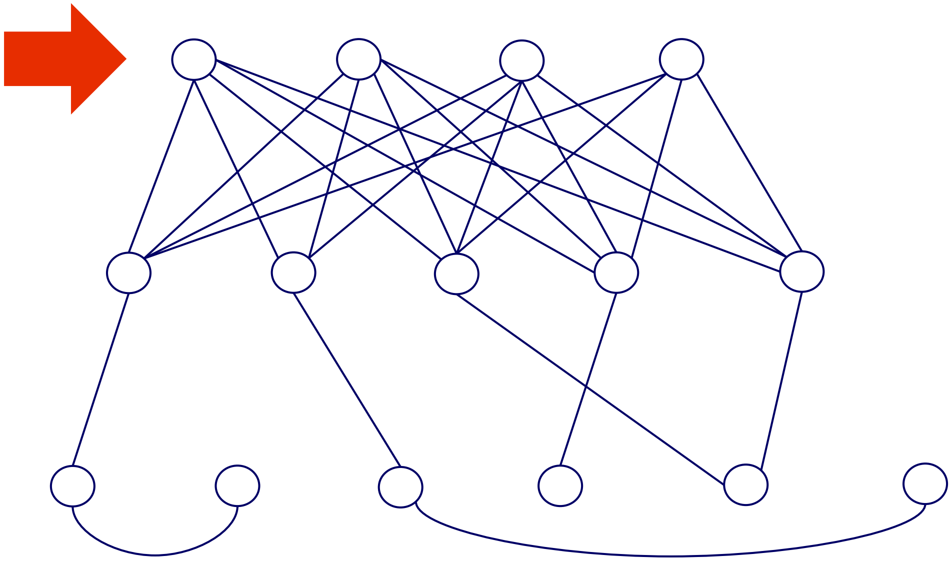


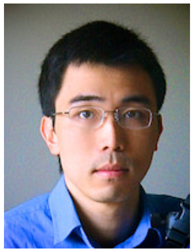
w/ Polo Chau &
Shashank Pandit, CMU
[www'07]



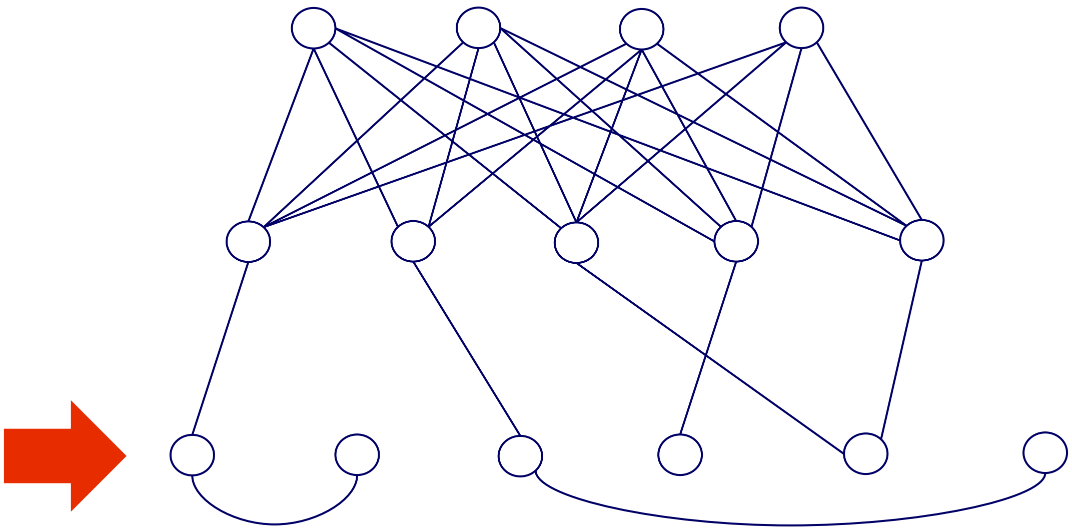
E-bay Fraud detection



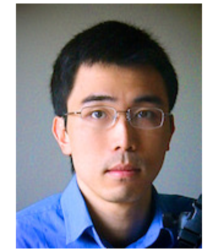


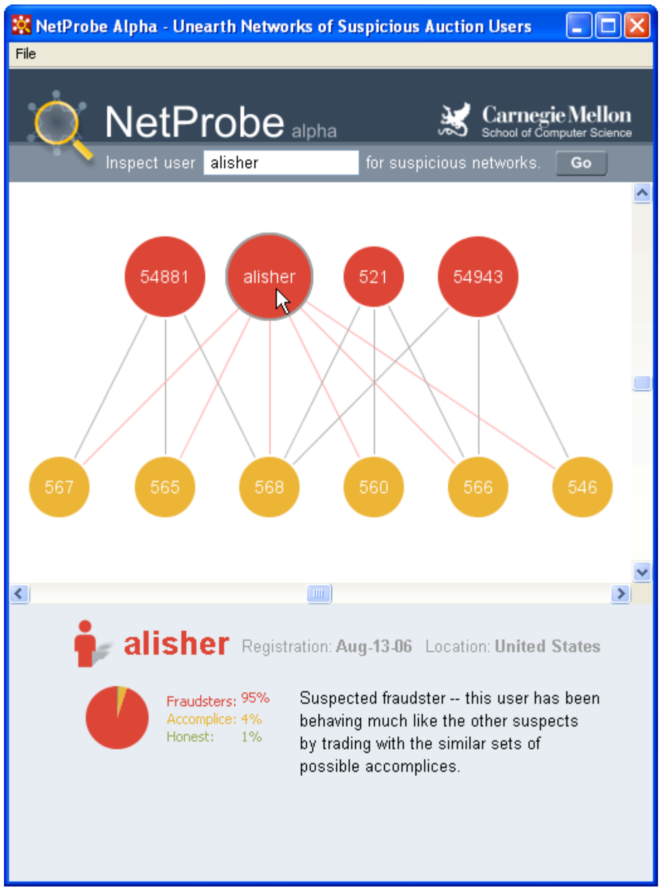


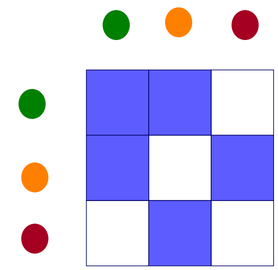
E-bay Fraud detection

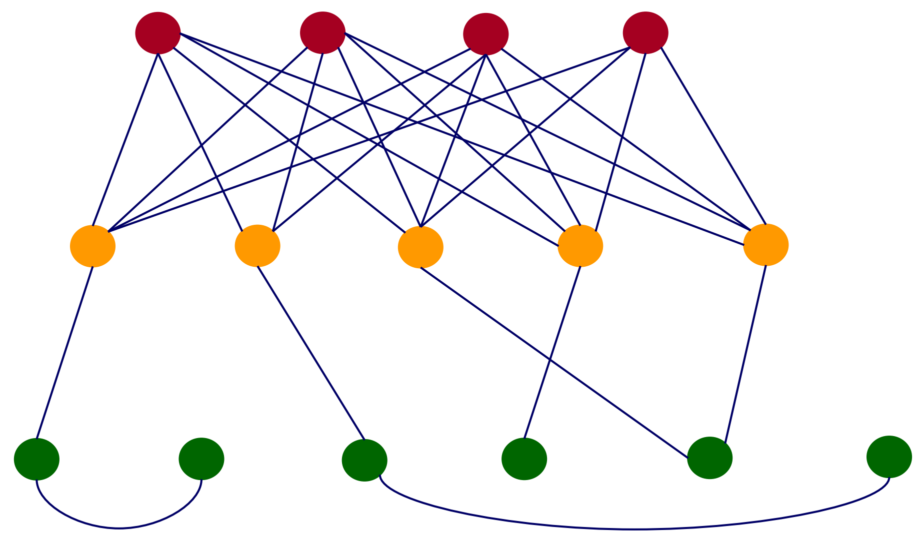


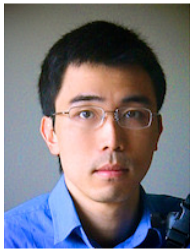
E-bay Fraud detection - NetProbe











Popular press

USA TODAY

The Washington Post

Los Angeles Times

And less desirable attention:

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

Roadmap

- Introduction – Motivation
- Unsupervised
- Supervised: BP (Belief Propagation)
  - Ebay fraud (‘NetProbe’)
  - Malware (Polonium)
  - Intuition behind BP
- Conclusions

Carnegie Mellon

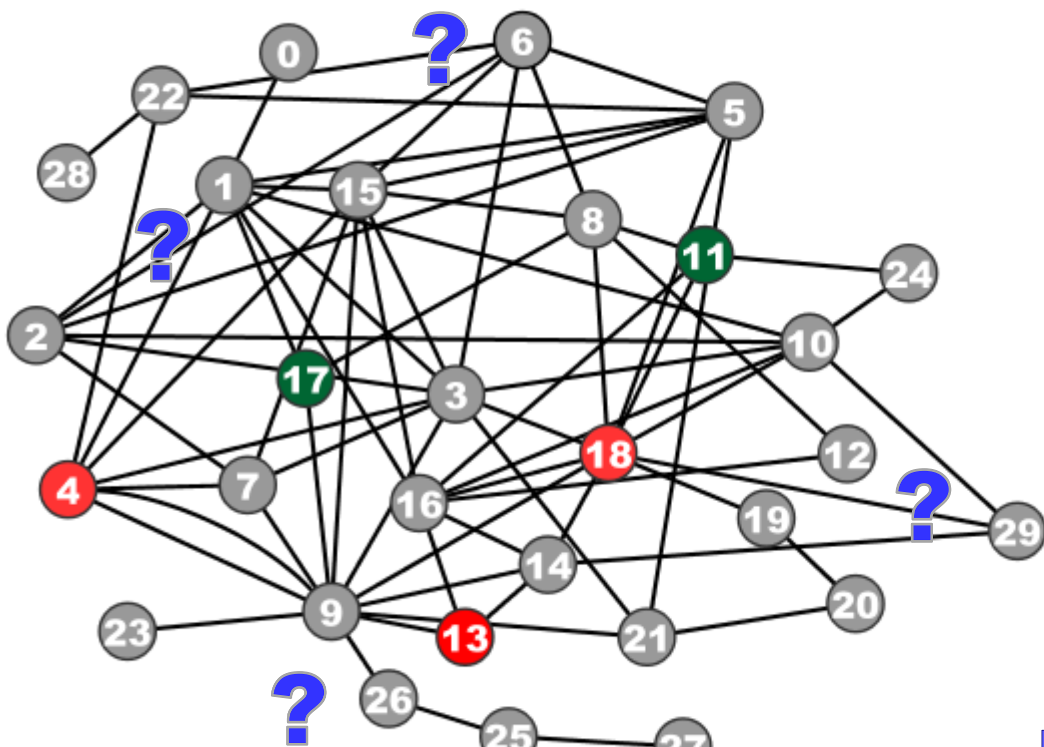
Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms

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

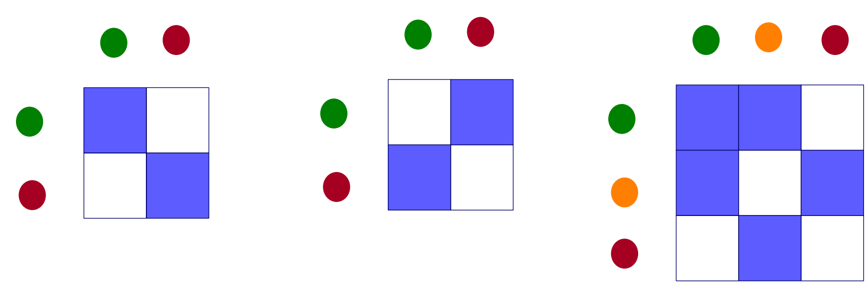
Carnegie Mellon

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	$[I - c\,\overline{A}\,\underline{D}^{-1}]$	$\times$	x	$=$	$(1-c)y$
SSL	$[I + a(D - \underline{A})]$	$\times$	x	$=$	y
FaBP	$[I + a D - c' A]$	$\times$	b_h	$=$	φ_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

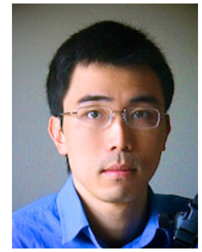
prior labels/ beliefs

Cast

Akoglu,
Leman

Araujo,
Miguel

Beutel,
Alex



Chau,
Polo

Kang, U

Koutra,
Danai

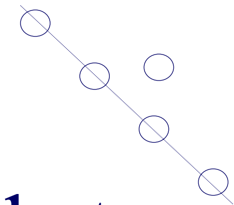
Lee,
Jay Yoon

Prakash,
Aditya

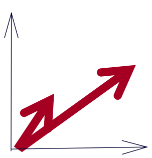
Papalexakis,
Vagelis

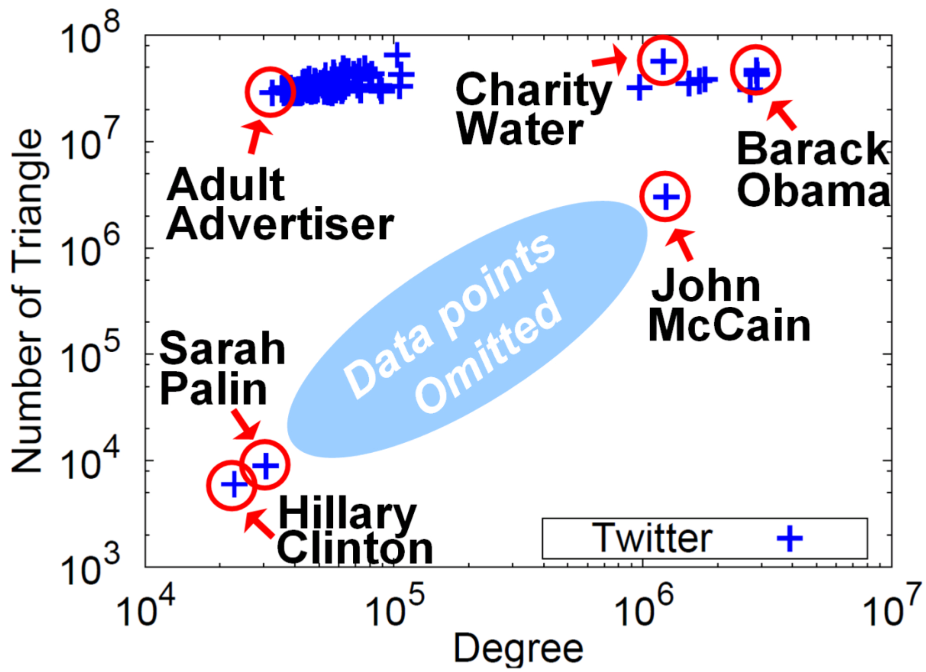
Shah,
Neil

Insights:



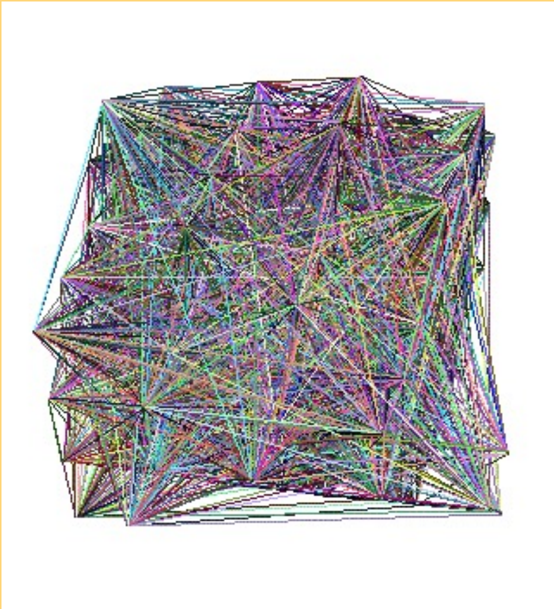
- **Patterns**  **Anomalies**
- **Large** datasets reveal patterns/outliers that are invisible otherwise





Solutions

Given:

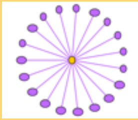


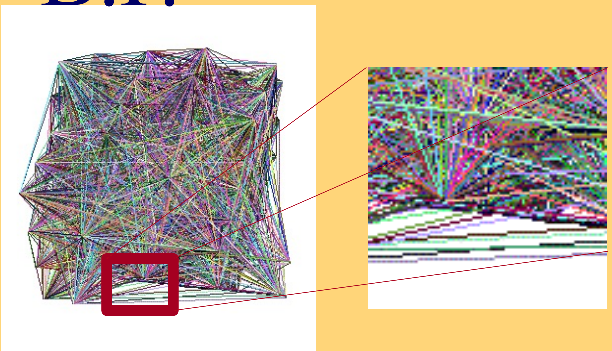
Find:

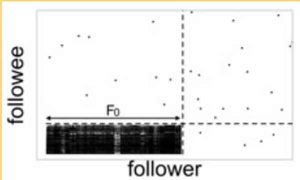
1) Outliers
2) Lock-step
3) B.P.

OddBall

SVD



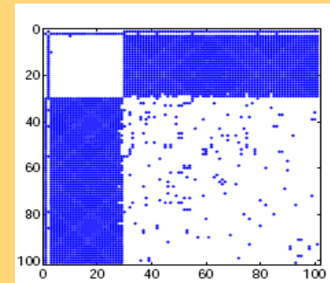




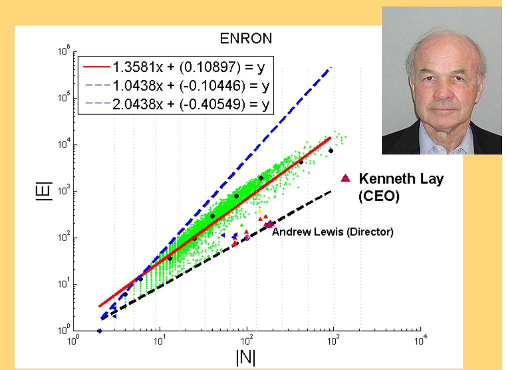
Solutions, cont'd

un-supervised

- Spectral/tensor methods: spot lockstep behavior



- OddBall: strange, single nodes



semi-supervised

- BP: good for the supervised setting (ie., some labels are given)