KDD 2017
Halifax, Nova Scotia - Canada
August 13 - 17, 2017

Tutorial: Data-Driven Approaches towards Malicious Behavior Modeling

Meng Jiang
University of Notre Dame

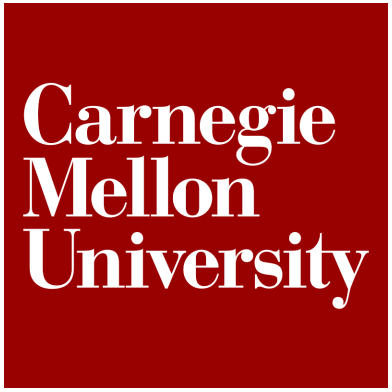
Srijan Kumar
Stanford University

Christos Faloutsos
Carnegie Mellon University

V.S. Subrahmanian
University of Maryland, College Park

http://www.meng-jiang.com/tutorial-kdd17.html

Acknowledgement



Outline

Introduction

Feature-based algorithms

- Bots
- Sockpuppets
- Vandals
- Hoaxes

Spectral-based algorithms

- Visualization: “spokes”, “blocks”, “staircases”
- Camouflage
- Theoretical guarantee

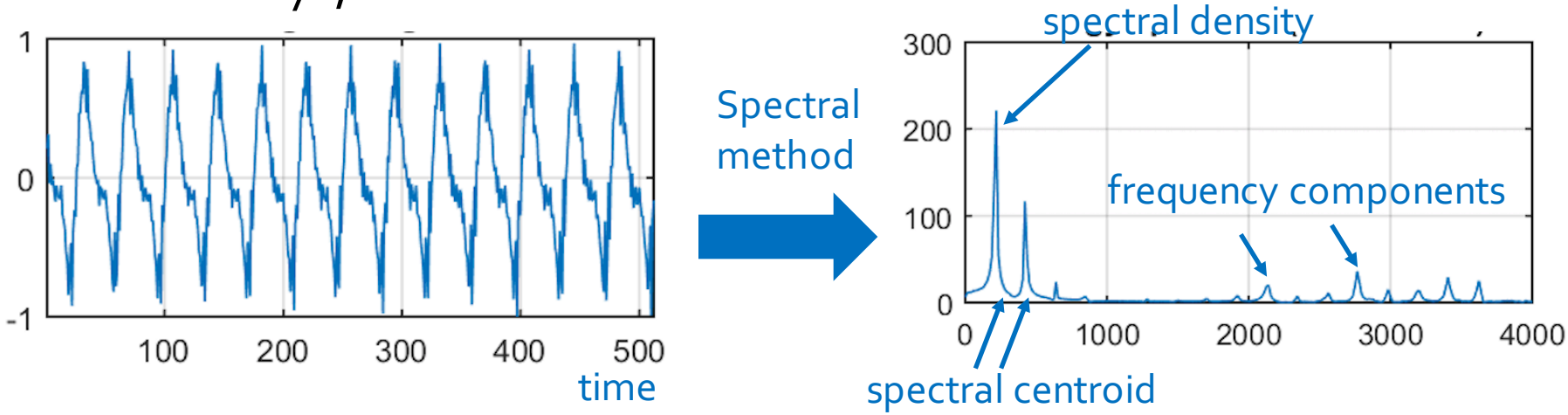
Density-based algorithms

- Ill-gotten Likes
- Synchronized Behaviors
- Advertising campaigns
- Social spam

Conclusions and future directions

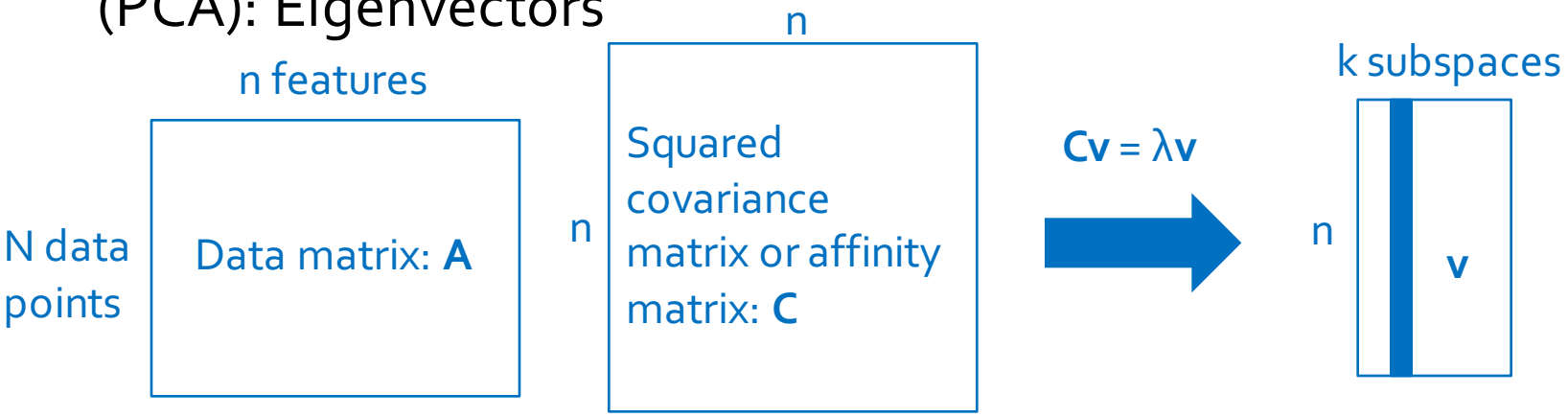
Spectral Features in Speech Signal Processing

- **Temporal features** (time domain features): which are simple to extract and have easy physical interpretation
- **Spectral features** (*frequency* based features): which are obtained by converting the time based signal into the frequency domain using the Fourier Transform, like "frequency components", "spectral centroid", "spectral density", etc.

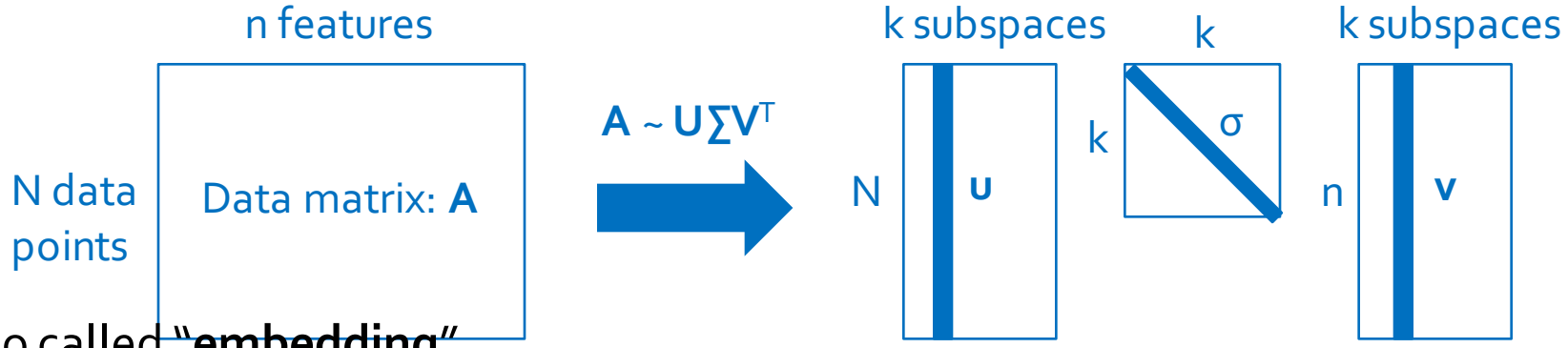


Spectral Subspaces

- Frequency components → Principal Component Analysis (PCA): Eigenvectors

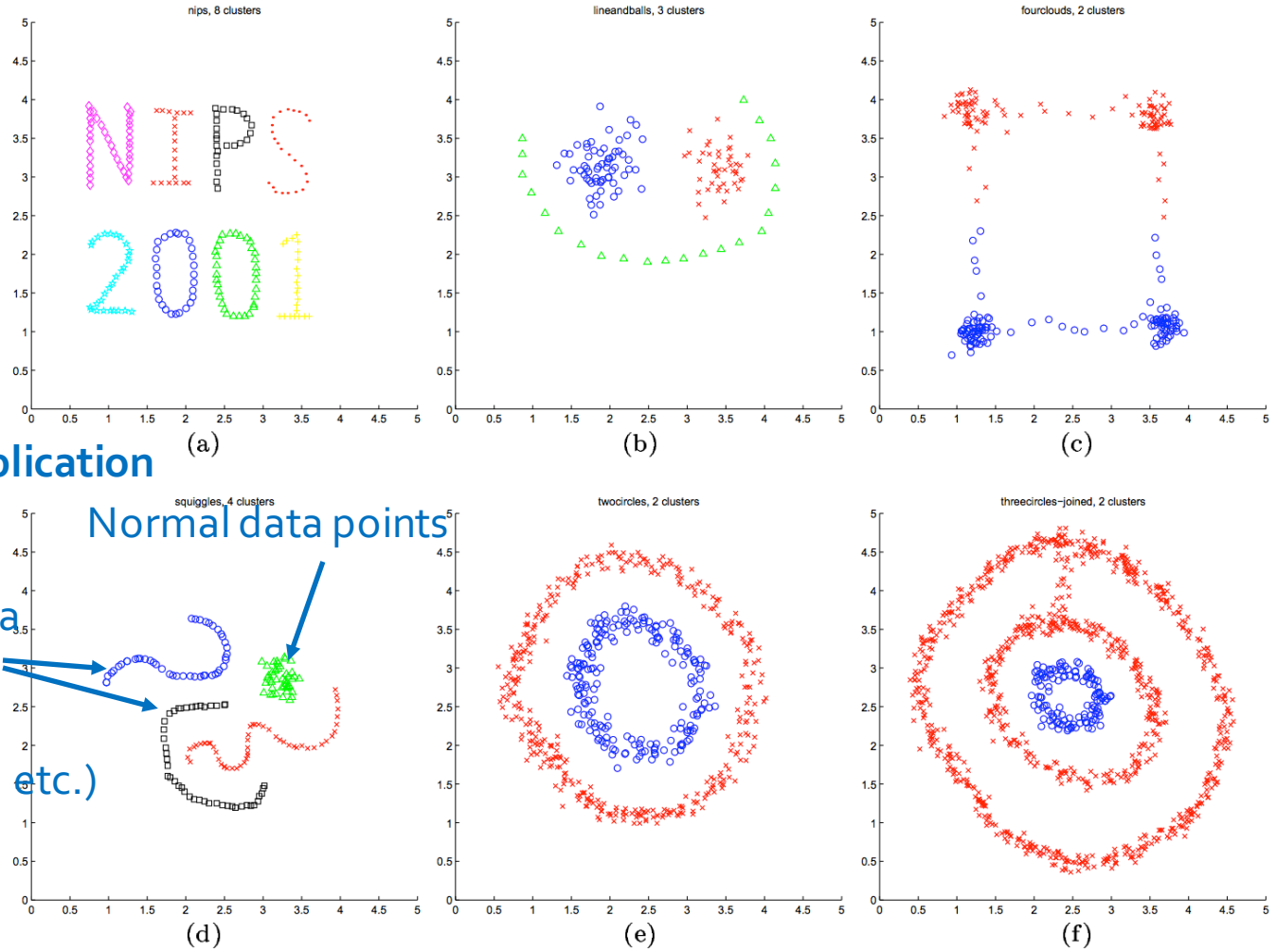


- Other spectral subspace methods. Singular Value Decomposition (SVD): Singular vectors



also called "**embedding**"

Spectral Clustering



Potential application

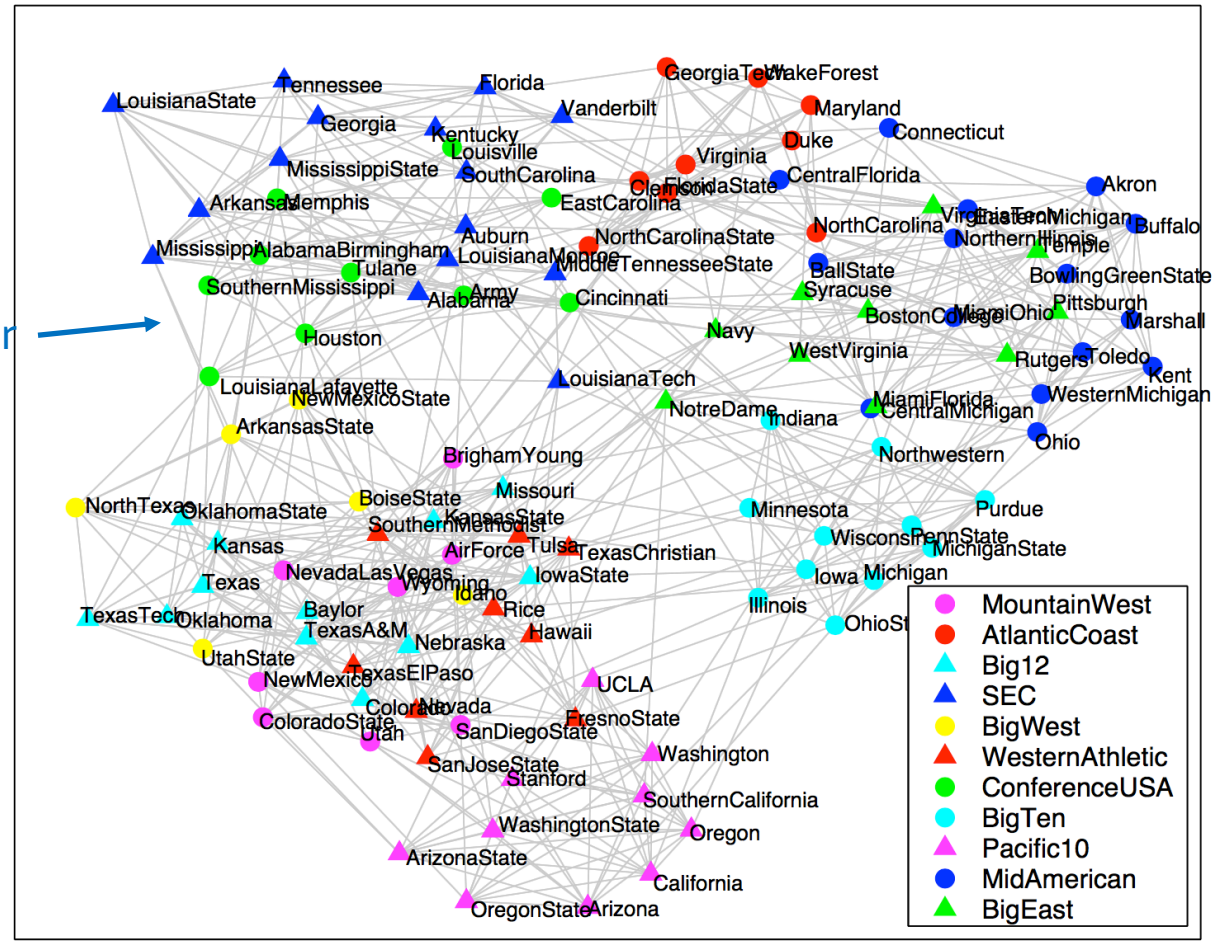
Normal data points

Malicious data points (users, behaviors, communities, etc.)

Andrew Y. Ng, Michael I. Jordan, Yair Weiss. "On Spectral Clustering: Analysis and an algorithm", NIPS 2001.

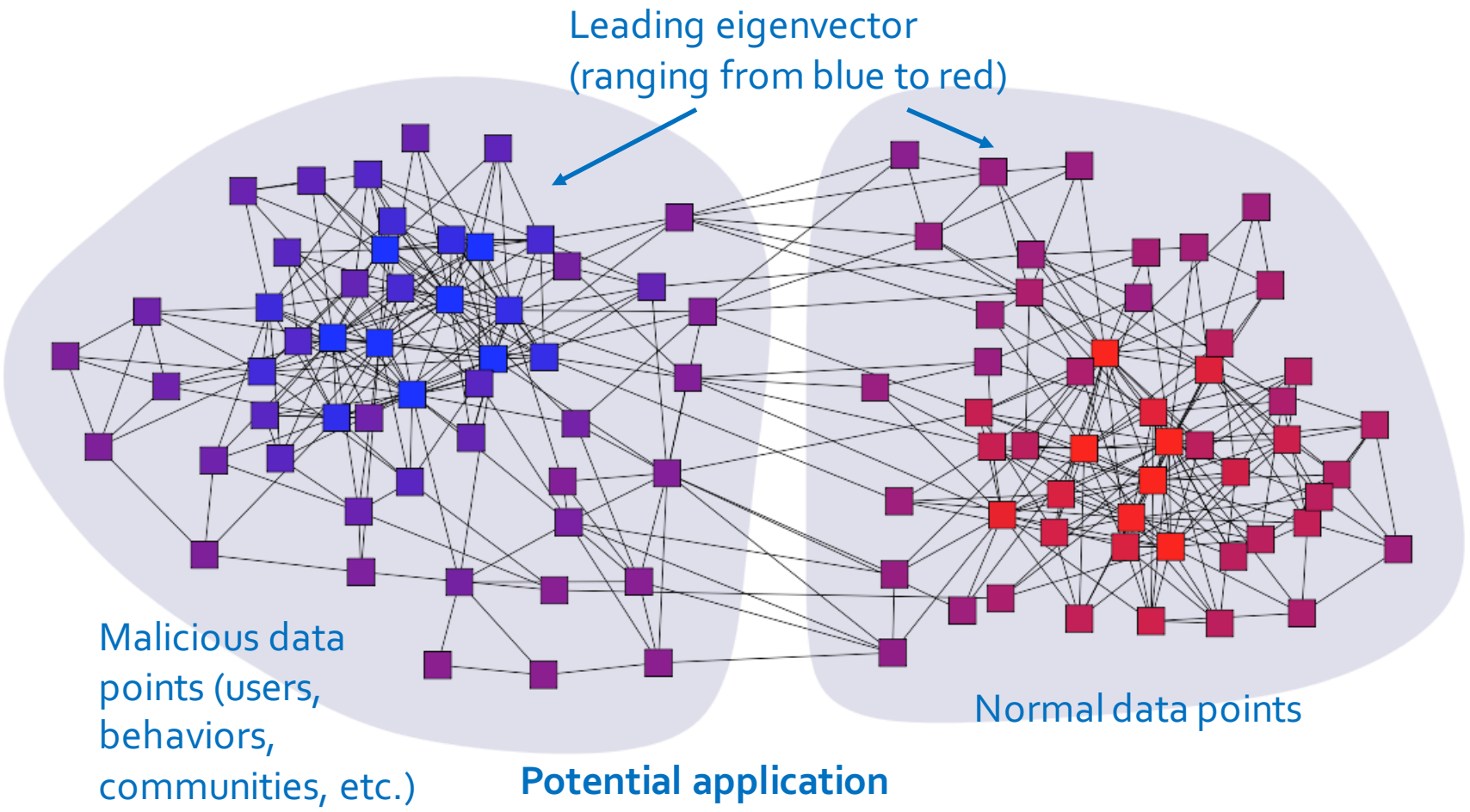
Spectral Methods for Communities

Played together



Scott White and Padhraic Smyth. "A spectral clustering approach to finding communities in graphs", SDM 2005.

Spectral Methods for Communities (cont.)



M. E. J. Newman. "Finding community structure in networks using the eigenvectors of matrices", Physical Review E 2006.

Spectral-based vs. Feature-based

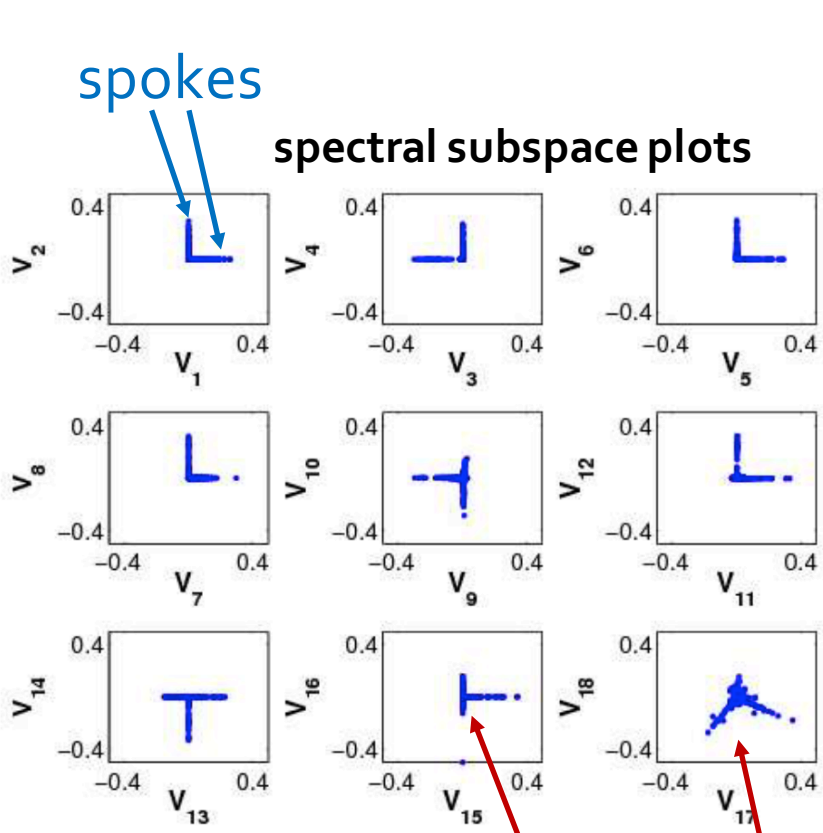
- Advantages
  - Visualization: k = Number of subspaces
  - Feature extraction by data distribution vs. selection
  - "Principal" components: "leading" vectors
  - Data: N-by-N graphs, N-by-N-by-N tensors
  - Applications: Finding communities and anomalies
- Disadvantages
  - Lack of interpretability of the subspaces/features

Finding Surprising Patterns in Large Graphs

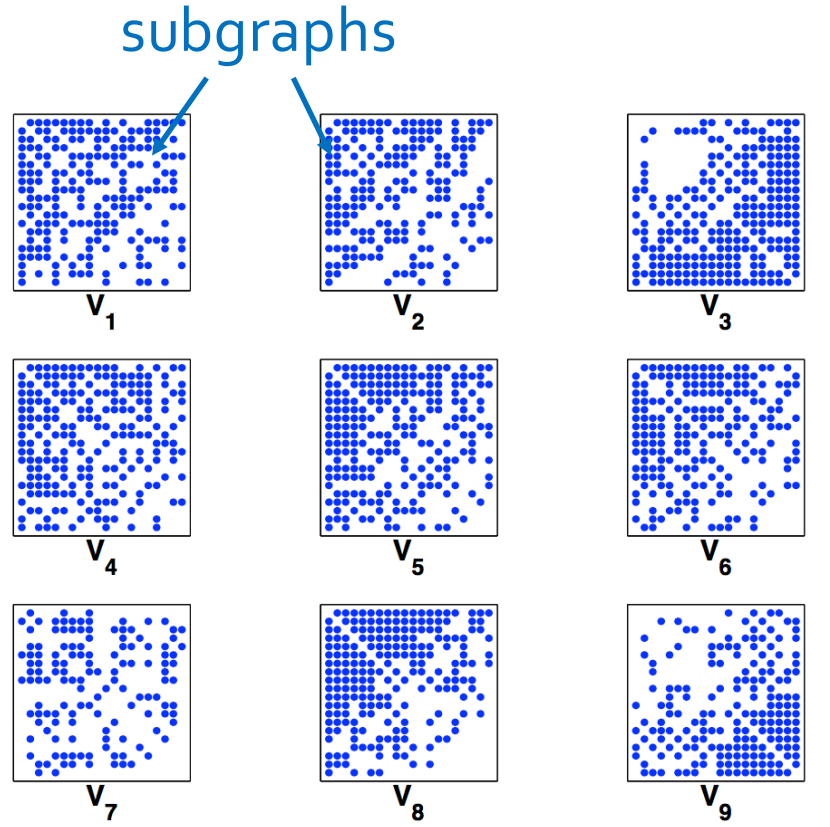
- Problem definition: Given a social graph based on mobile calls made from/to callers, find caller communities.
- Dataset: Activity over the duration of a month, 186,000 nodes and 464,000 edges.
- Key contribution: Discovery of the "spokes" phenomenon
  - **The singular vectors of the graph, when plotted against each other, often have clear separate lines, typically aligned with axes.**
  - Use EigenEigen (EE) plots to identify communities in the form of **cliques or near-cliques**, **perfect or near-perfect bipartite-cores**. (*Malicious?)

Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju, Christos Faloutsos. "EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs", PAKDD 2010.

Spokes and Dense Cliques



(a) *EE-plot*



(b) Spy Plots of Sub-graph of Top 20 Nodes

long vs. short?

tilted?

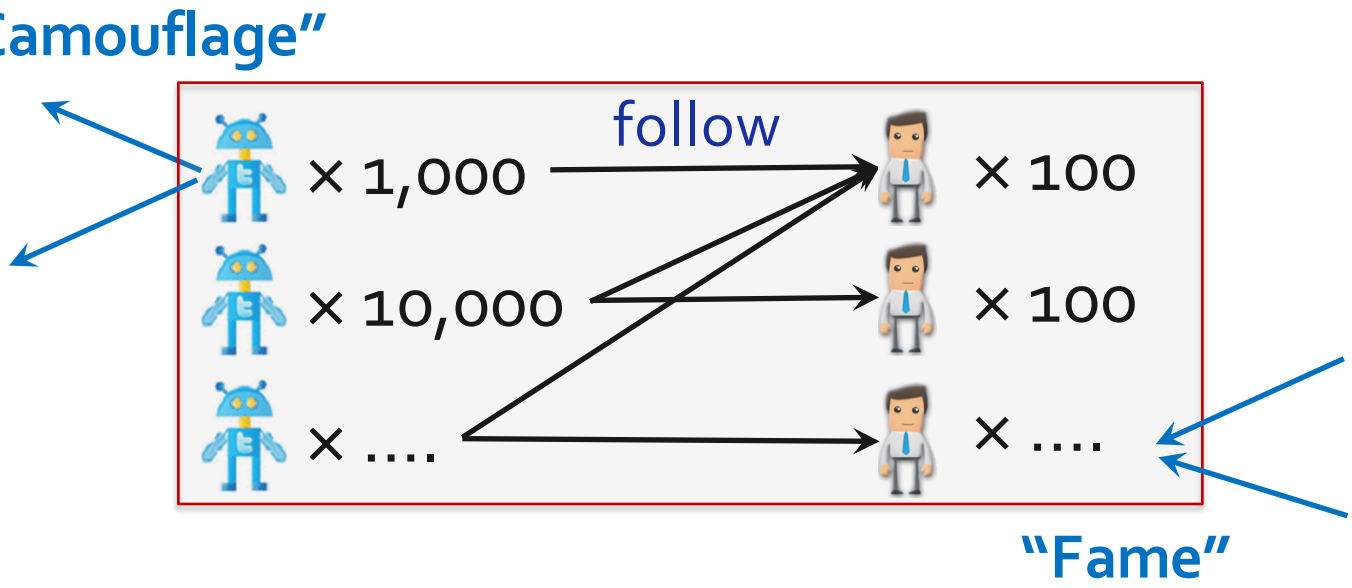
Inferring Lockstep Behavior from Connectivity Patterns

- Problem definition: Given a large graph, from spectral subspace plots, can we infer **lockstep behavior patterns**?

Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang. "Inferring lockstep behavior from connectivity pattern in large graphs", KAIS 2016.

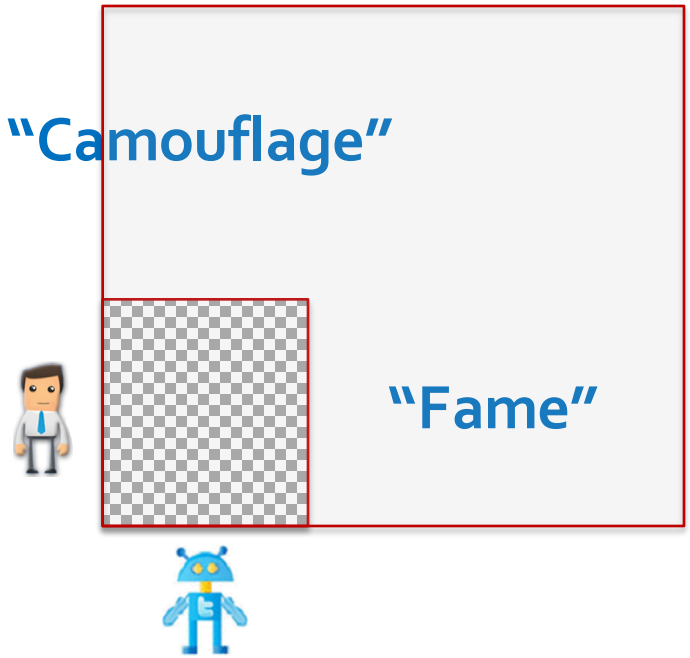
Inferring Lockstep Behavior from Connectivity Patterns

- Problem definition: Given a large graph, from **spectral subspace plots**, can we infer **lockstep behavior** patterns?



Inferring Lockstep Behavior from Connectivity Patterns

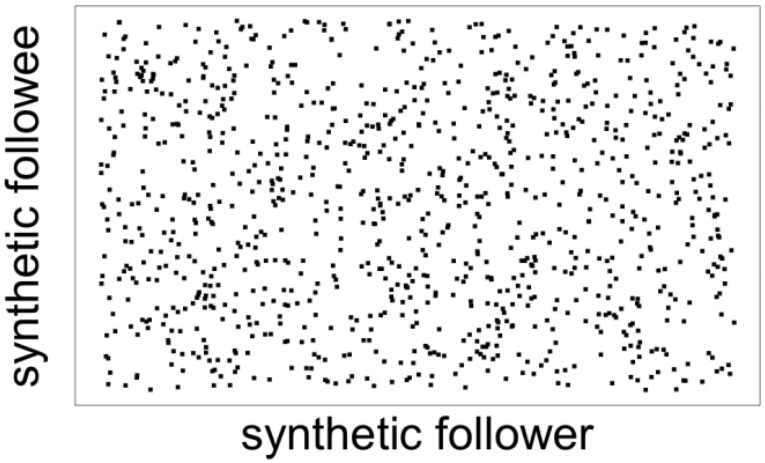
- Problem definition: Given a large graph, from **spectral subspace plots**, can we infer **lockstep behavior** patterns?



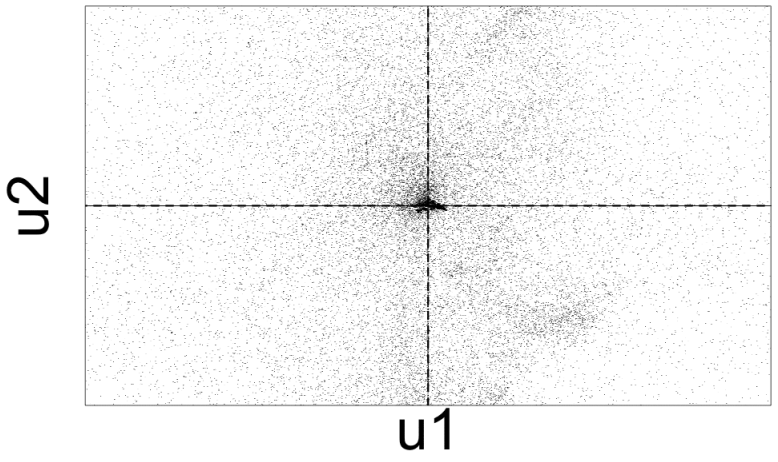
Case #0

- No lockstep behavior: Scatter

Adjacency Matrix



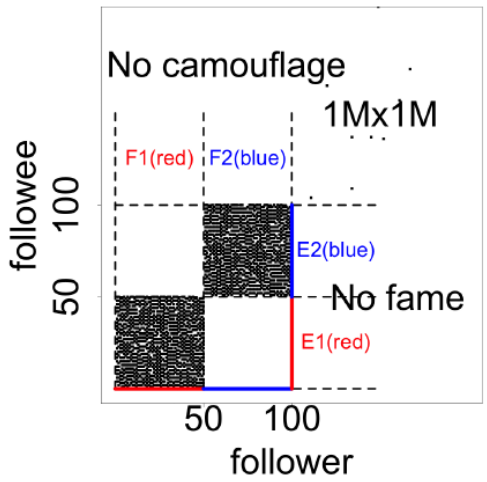
Spectral Subspace Plot



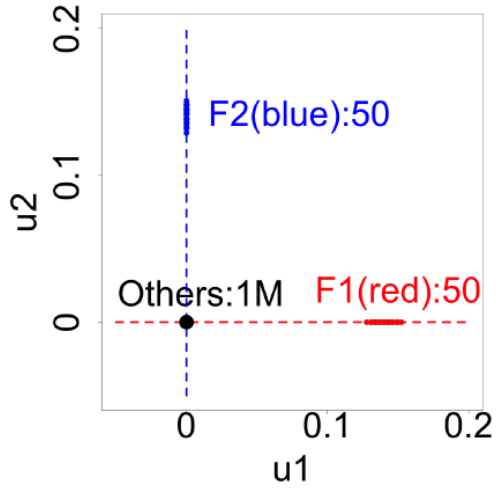
Case #1

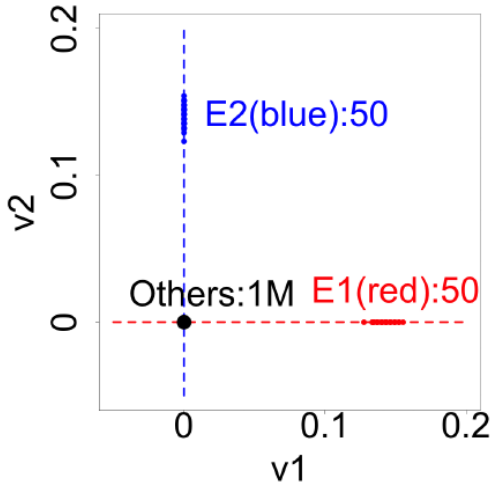
- Non-overlapping lockstep: "Rays"

Adjacency Matrix



Spectral Subspace Plot



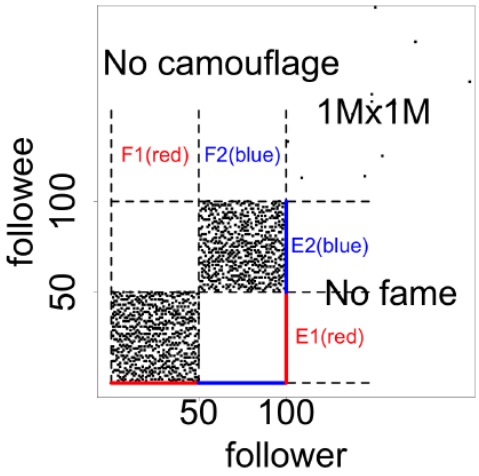


Rule 1 (short "rays"): two blocks, high density (90%), no "camouflage", no "fame"

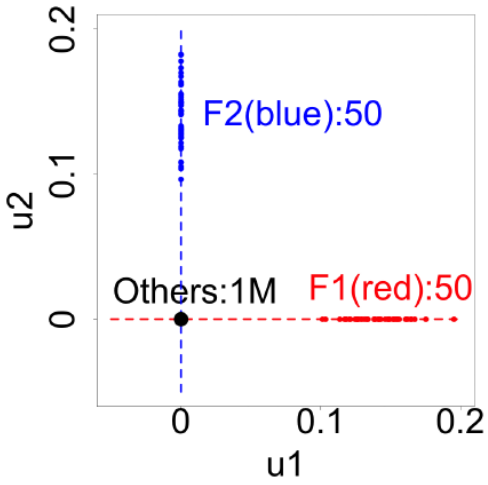
Case #2

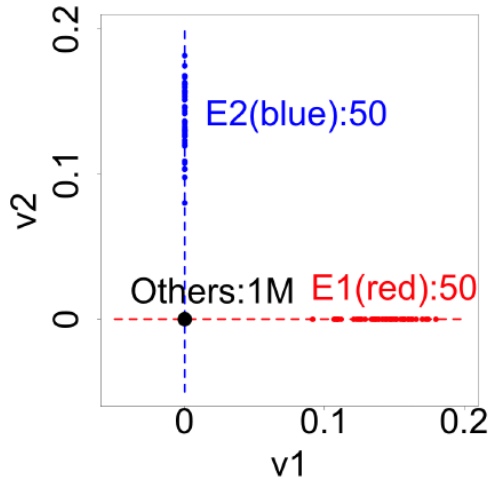
- Non-overlapping: Low density, Elongation

Adjacency Matrix



Spectral Subspace Plot



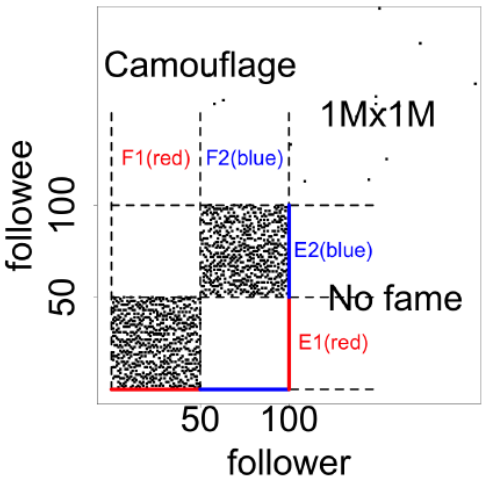


Rule 2 (long "rays"): two blocks, low density (50%), no "camouflage", no "fame"

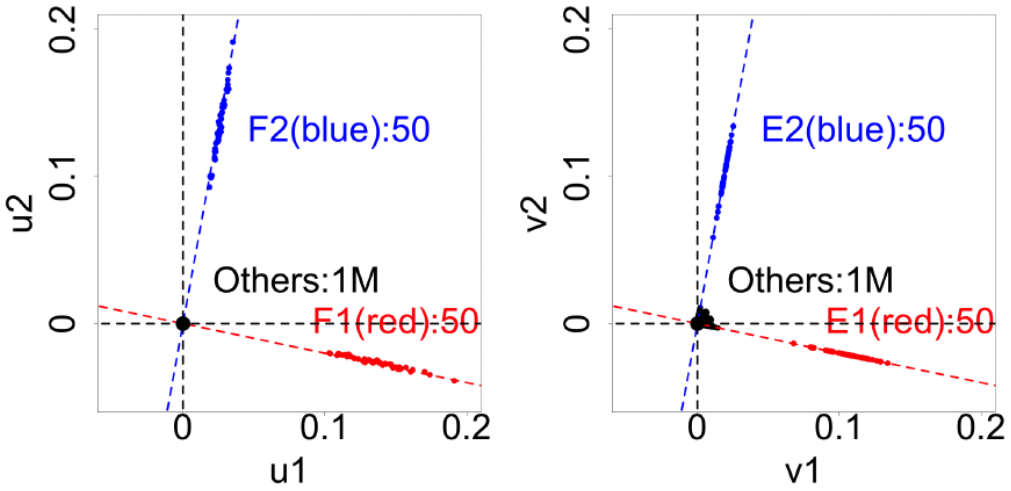
Case #3

- Non-overlapping: “Camouflage”/ “Fame”, Tilting

Adjacency Matrix



Spectral Subspace Plot

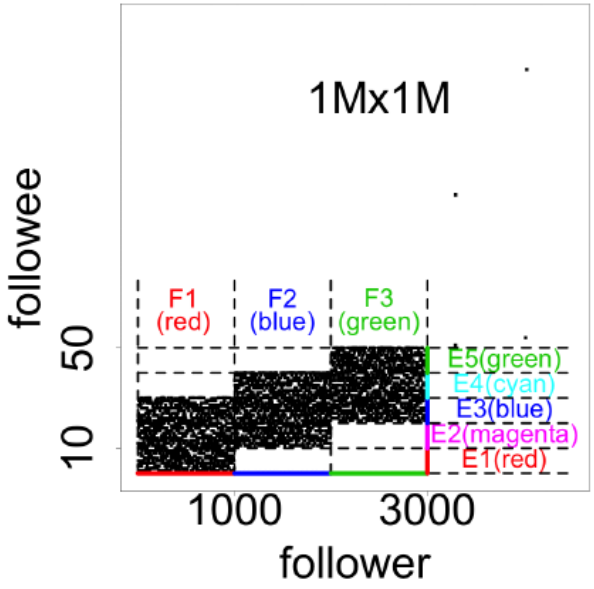


Rule 3 (tilting “rays”): two blocks, with “camouflage”, no “fame”

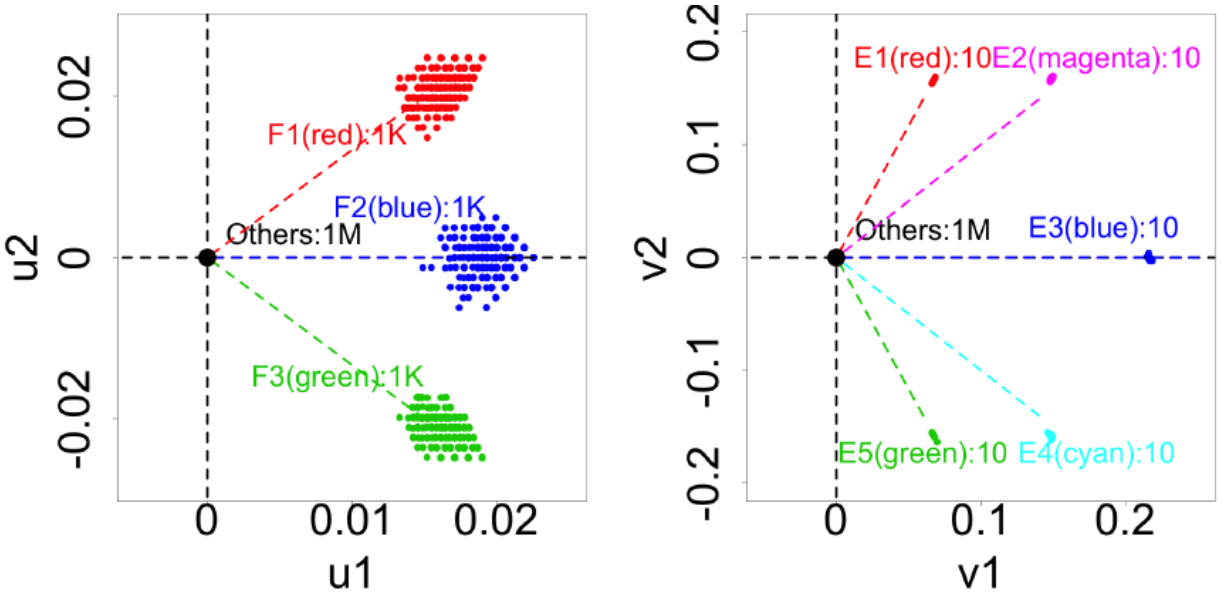
Case #4

- Overlapping: “Staircase”, “Pearls”

Adjacency Matrix



Spectral Subspace Plot



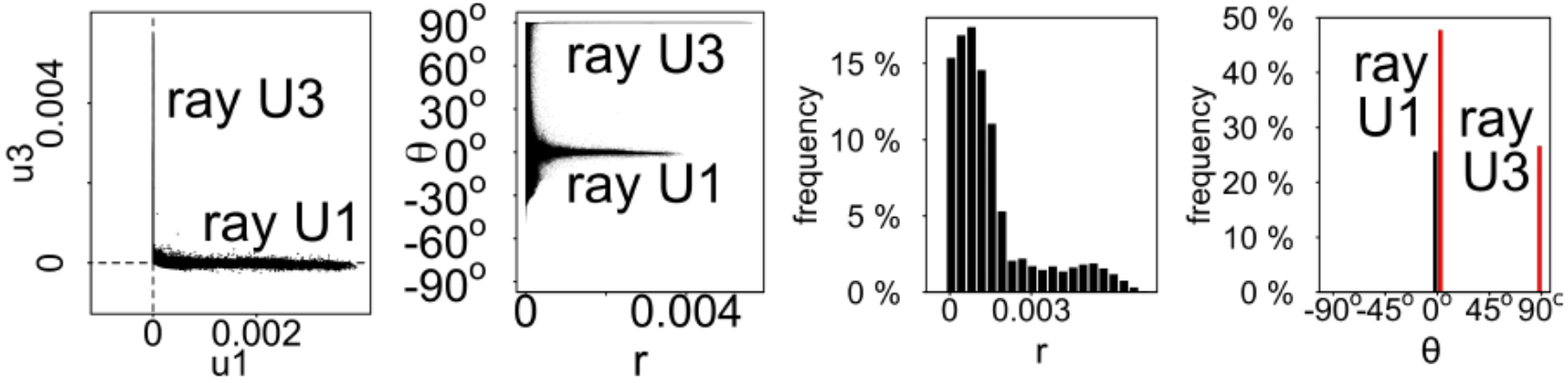
Rule 4 (“pearls”): a “staircase” of three partially overlapping blocks.

LockInfer Algorithm: Reading Spectral Subspace Plots

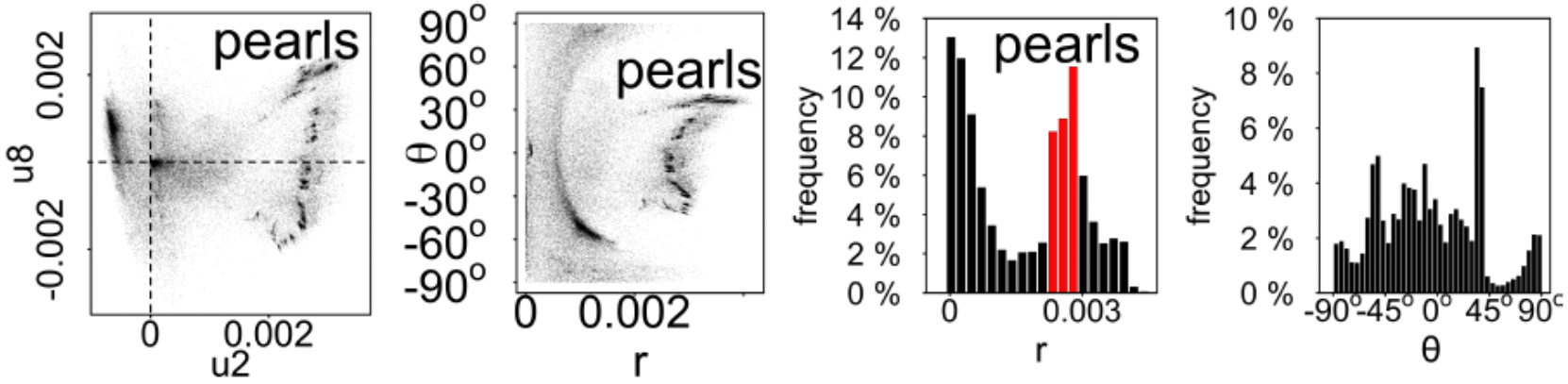
Spectral Subspace Plot

Polar Coordinate Transform

Histograms



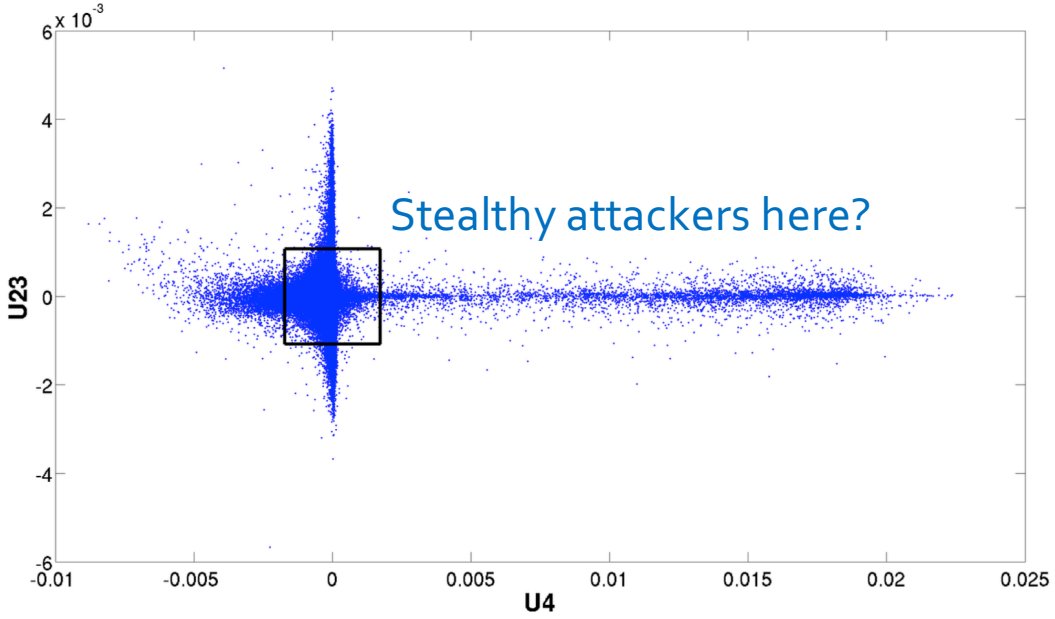
"rays" show two apparent spikes on θ frequency at 0^o and 90^o



"pearls" show a spike on r frequency at a much-greater-than-zero value

Spotting Small-Scale, Stealthy Attacks

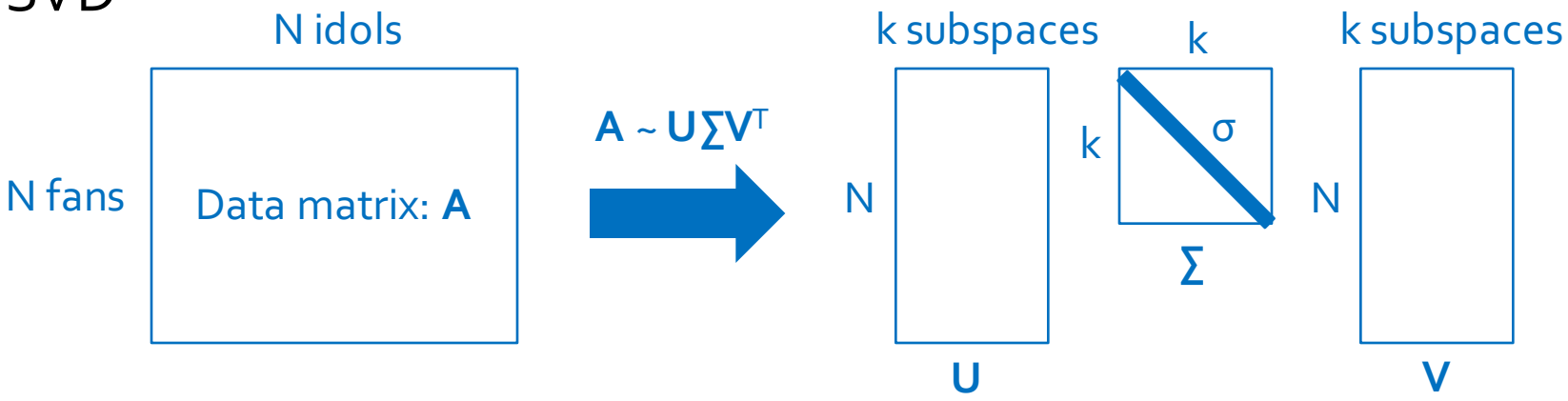
- Problem definition: Can we catches stealthy attacks which are missed by traditional spectral methods?
- Dataset: Twitter "who-follows-whom" social graph, 41.7 million nodes, 1.5 billion edges



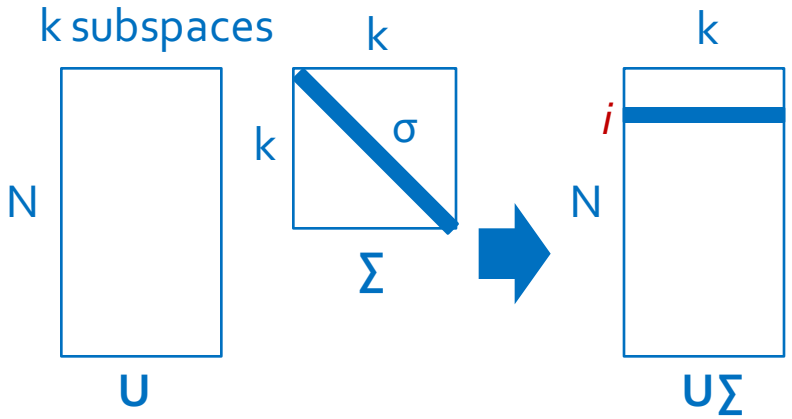
Neil Shah, Alex Beutel, Brian Gallagher, Christos Faloutsos. "Spotting suspicious link behavior with fBox: An adversarial perspective", ICDM 2014.

fBox: Reconstructed Degrees

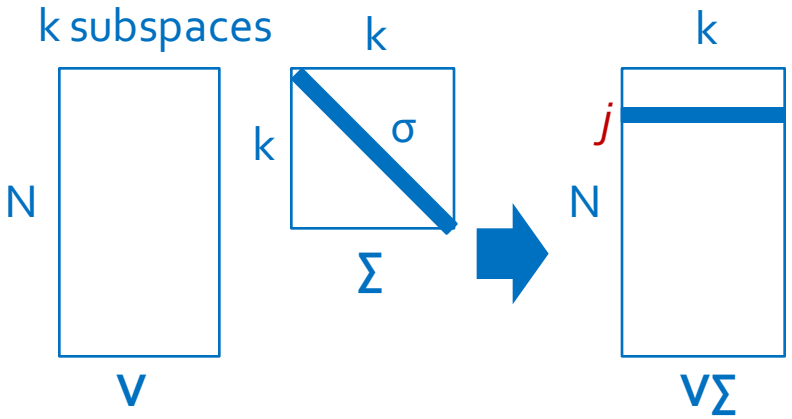
SVD



Reconstructed out-degree(i) = $||(\mathbf{U\Sigma})_i||_2$



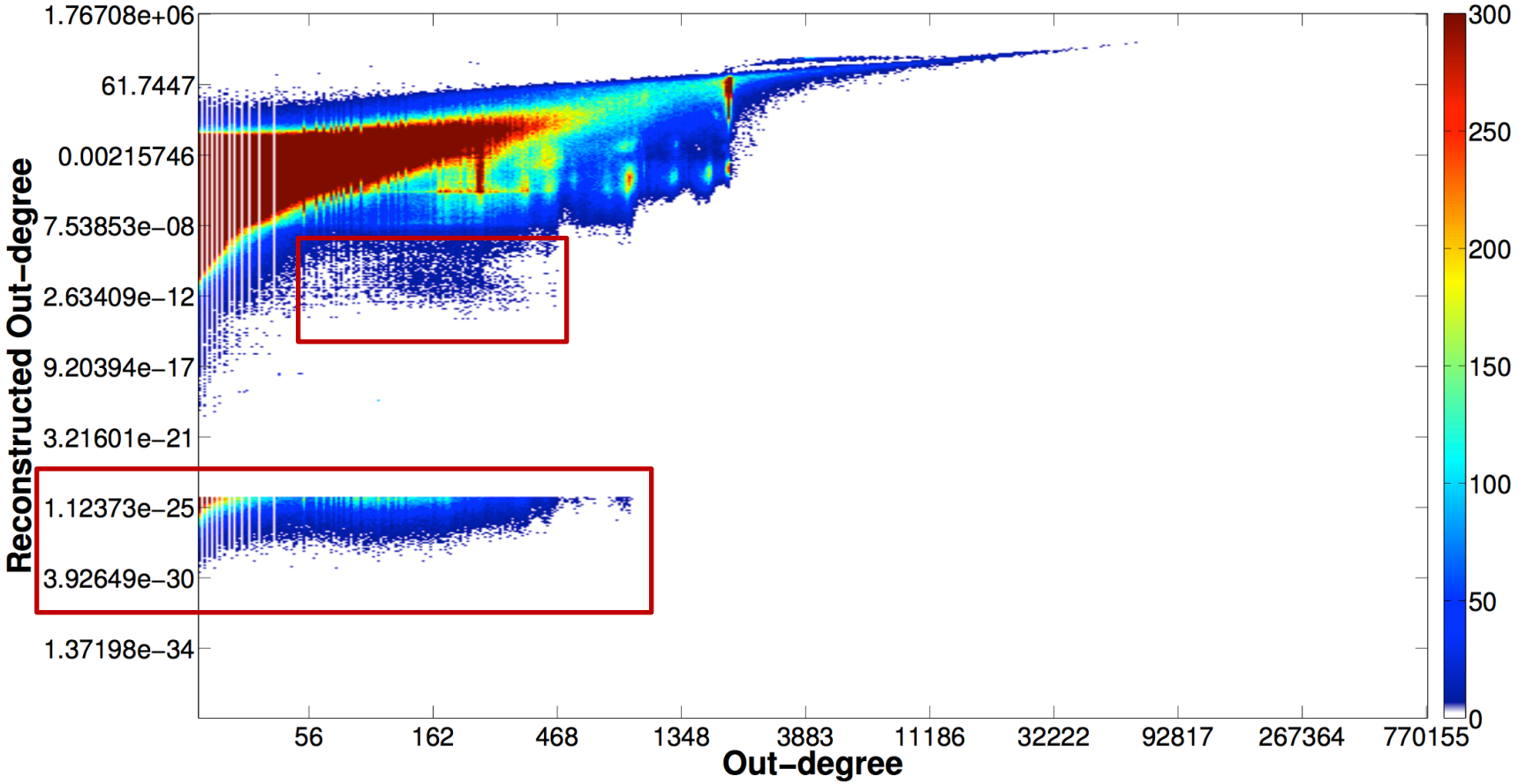
Reconstructed in-degree(j) = $||(\mathbf{V\Sigma})_j||_2$



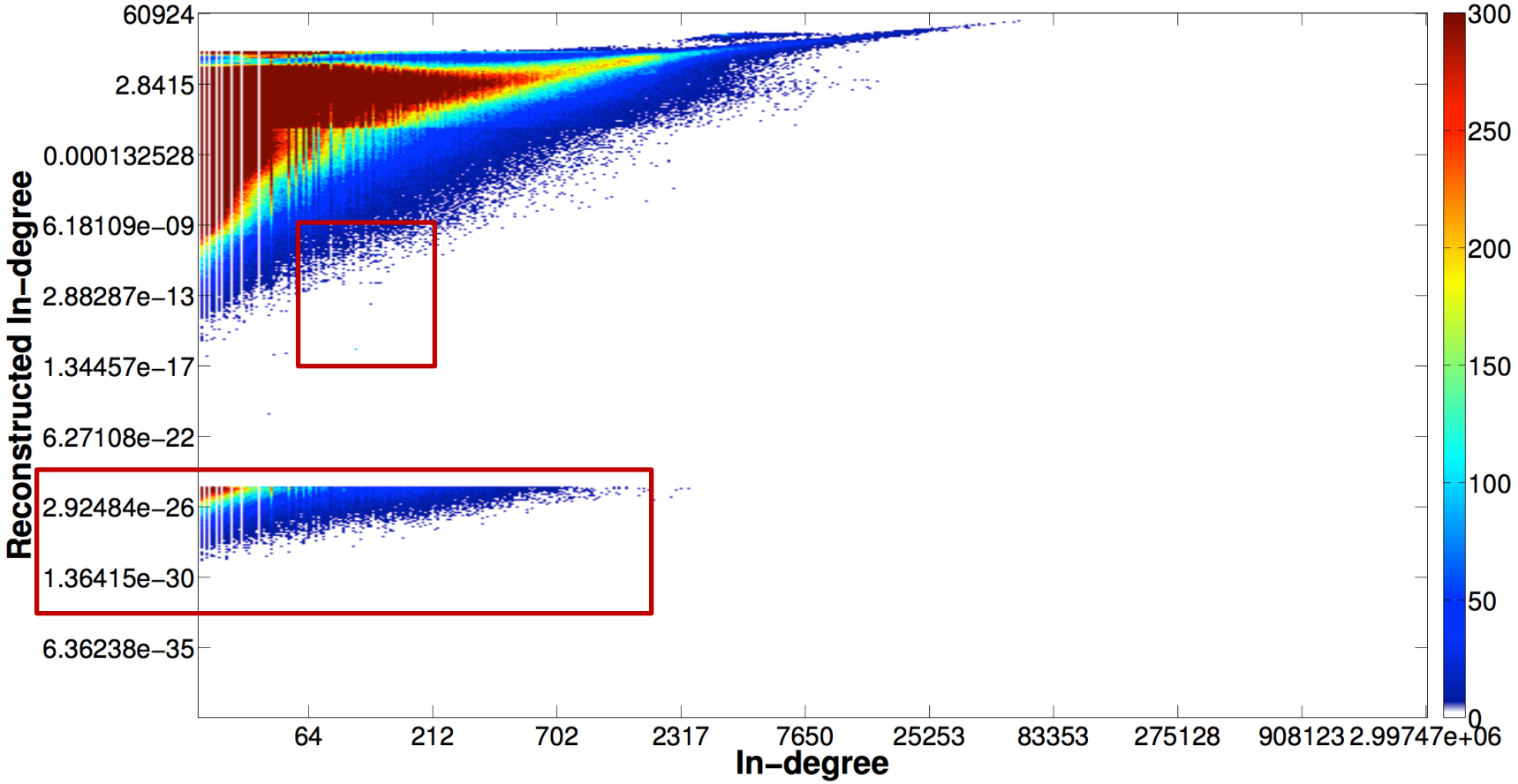
Norm-Preserving Property of SVD

- The row vectors of a full rank decomposition and associated projection will retain the same *l2* norm or vector length as in the original space:
  - For $k = \text{rank}(\mathbf{A})$, $\|\mathbf{A}_i\|_2 = \|(\mathbf{U\Sigma})_i\|_2$ and $\|\mathbf{A}^T_j\|_2 = \|(\mathbf{V\Sigma})_j\|_2$
- Plots
  - **Reconstructed out-degree vs. Real out-degree**
  - **Reconstructed in-degree vs. Real in-degree**
  - Can we find communities (groups of a number of users) whose reconstructed degrees are very small but in the same range?

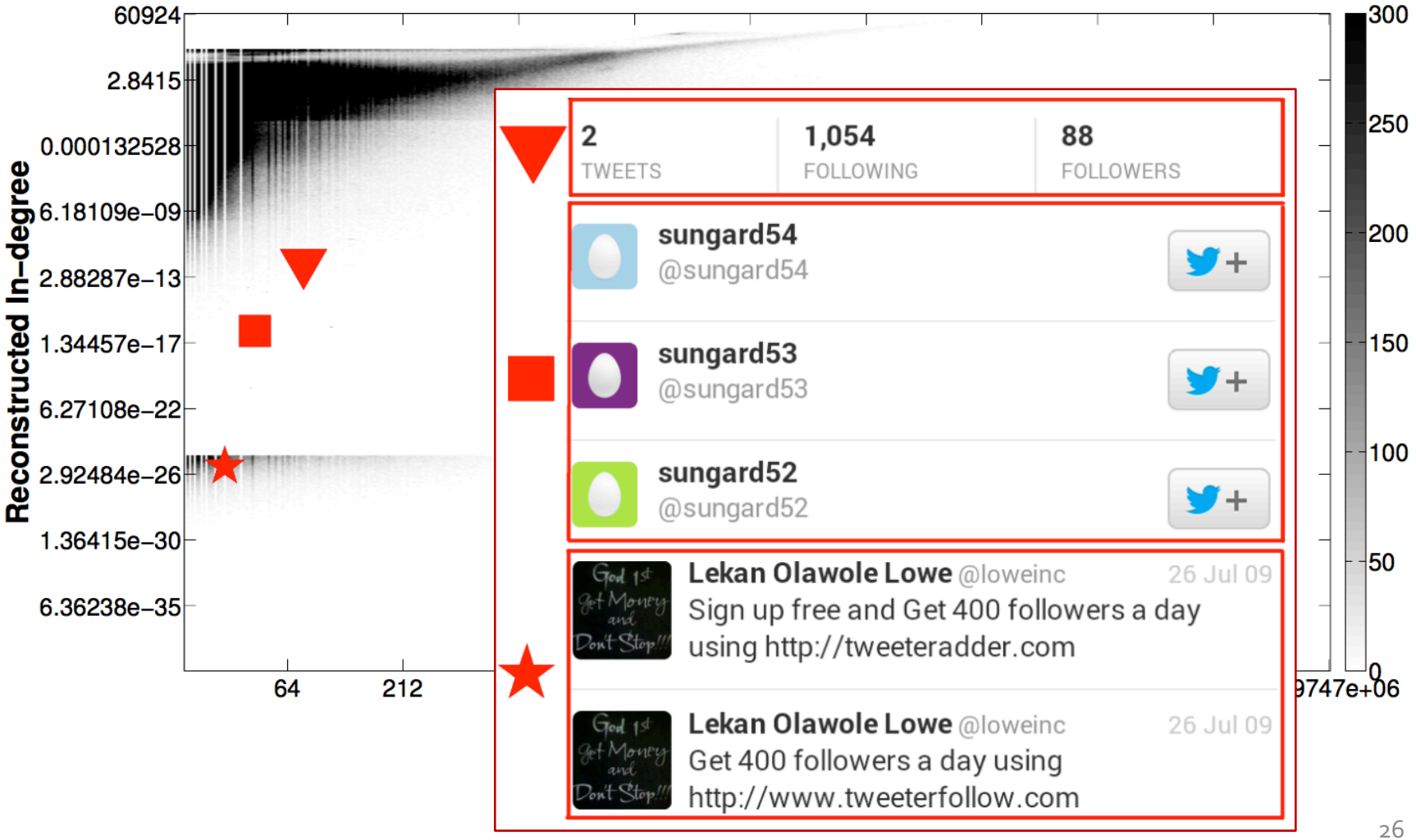
Reconstructed out-degree vs. Real out-degree



Reconstructed in-degree vs. Real in-degree

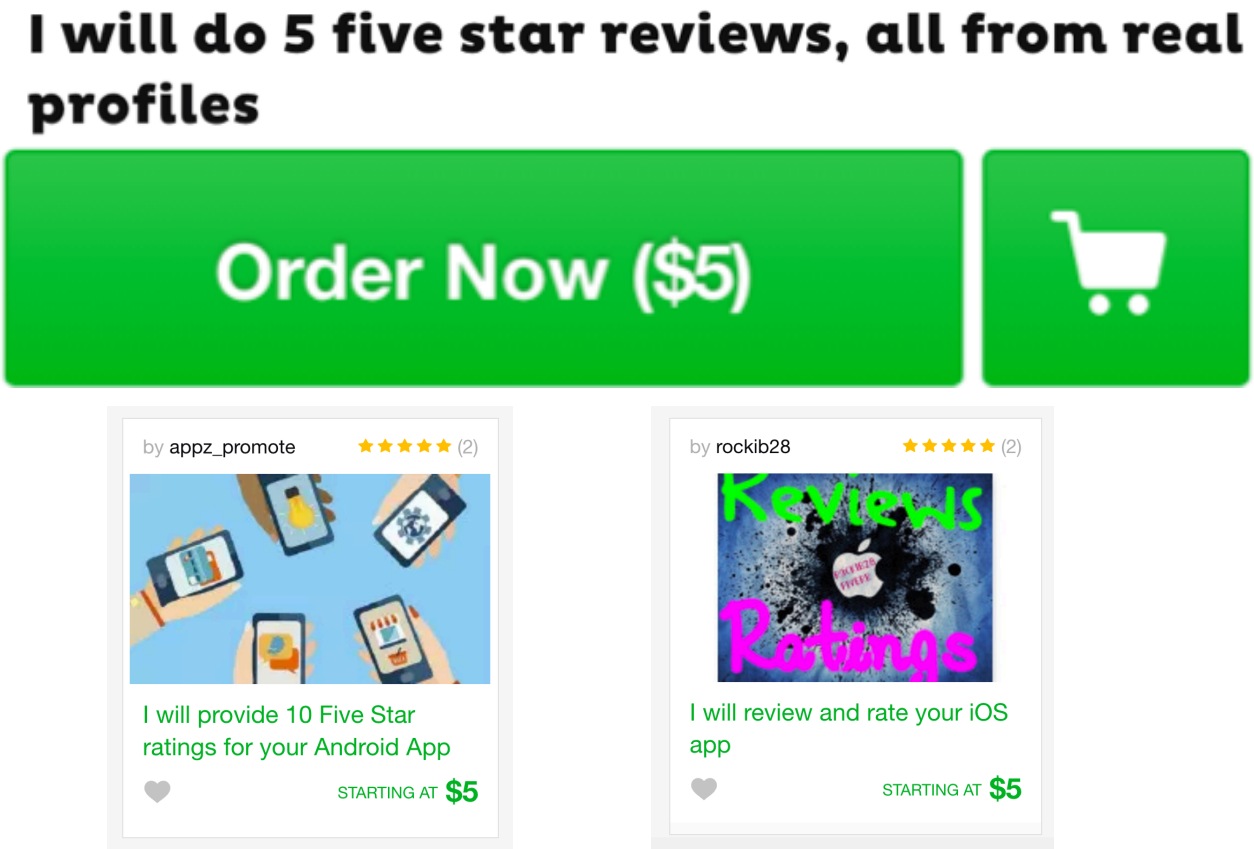


Reconstructed in-degree vs. Real in-degree (cont.)



Bounding Graph Fraud in Camouflage

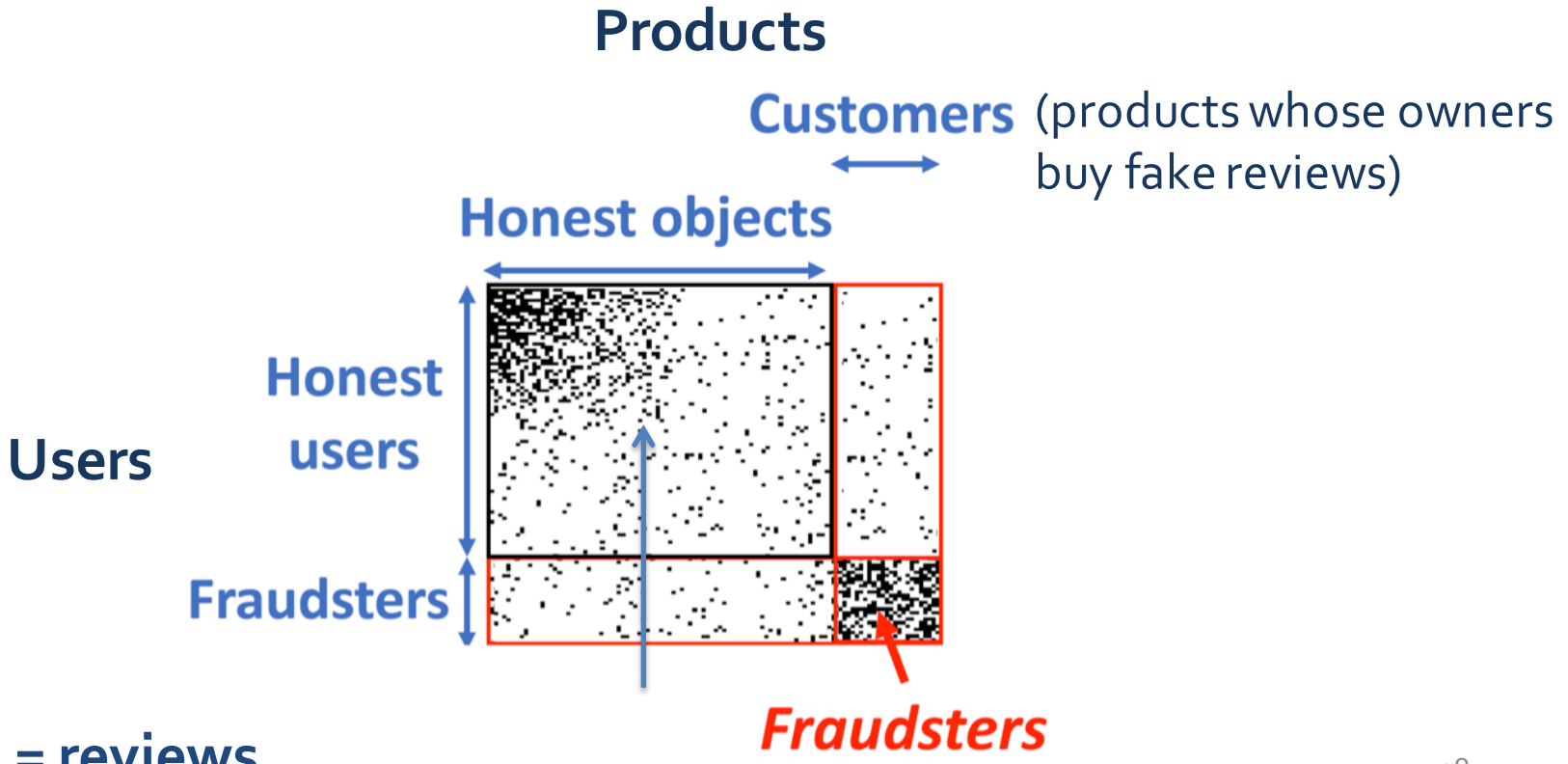
- One application: Fake reviews



Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, Christos Faloutsos. "FRAUDAR: Bounding Graph Fraud in the Face of Camouflage", **KDD 2016 Best Research Paper Award**.

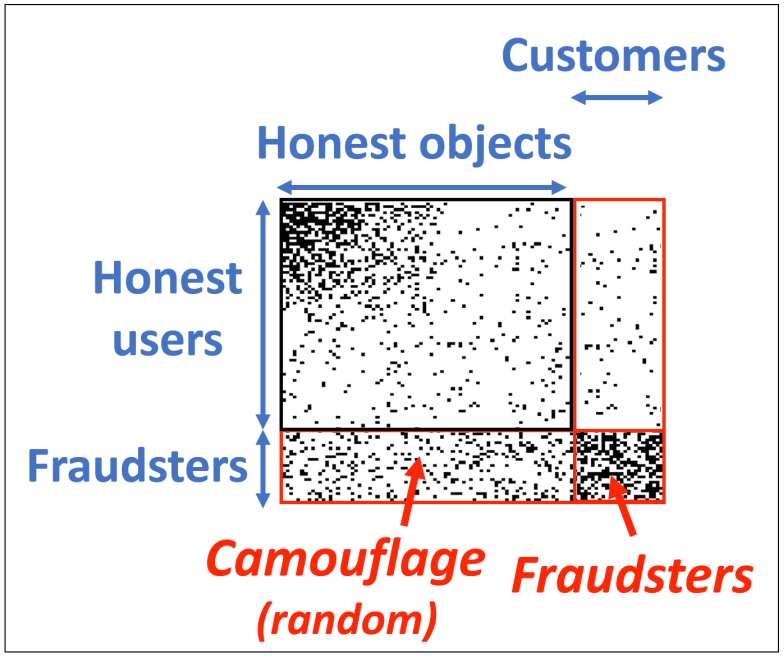
"User-Product" Review Graph

- Problem definition: Given a "user-product" review graph, can we spot fraudsters and customers?



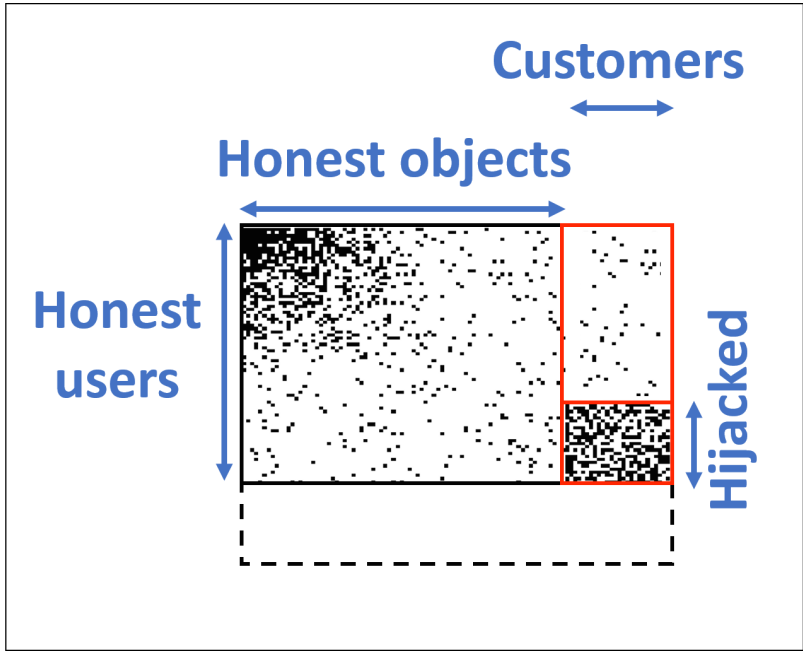
Camouflage: Evading Detection

Random camouflage



= "camouflage" in LockInfer

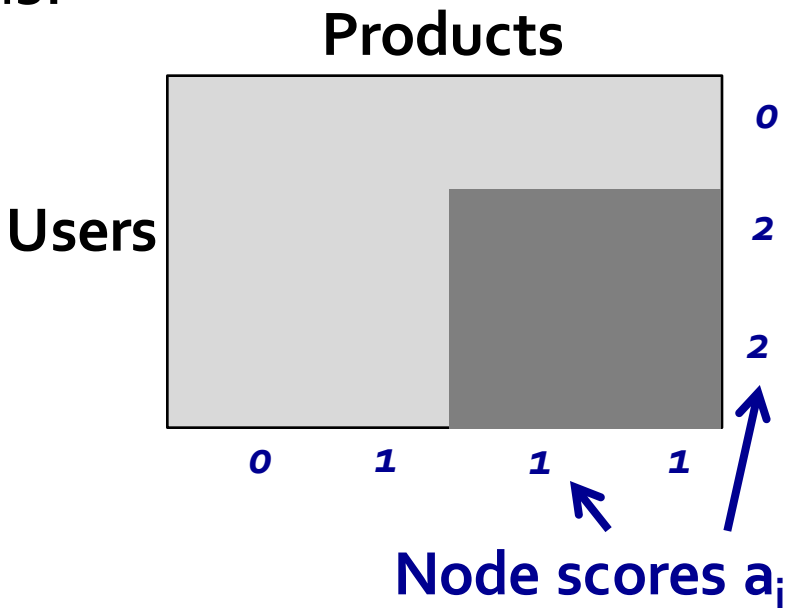
Hijacked user accounts



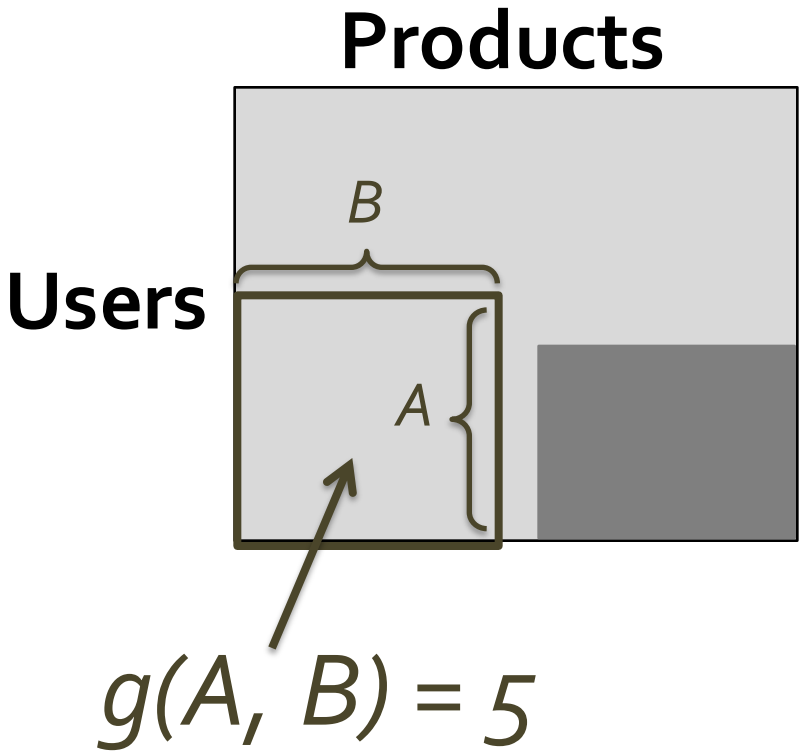
= "fame" in LockInfer

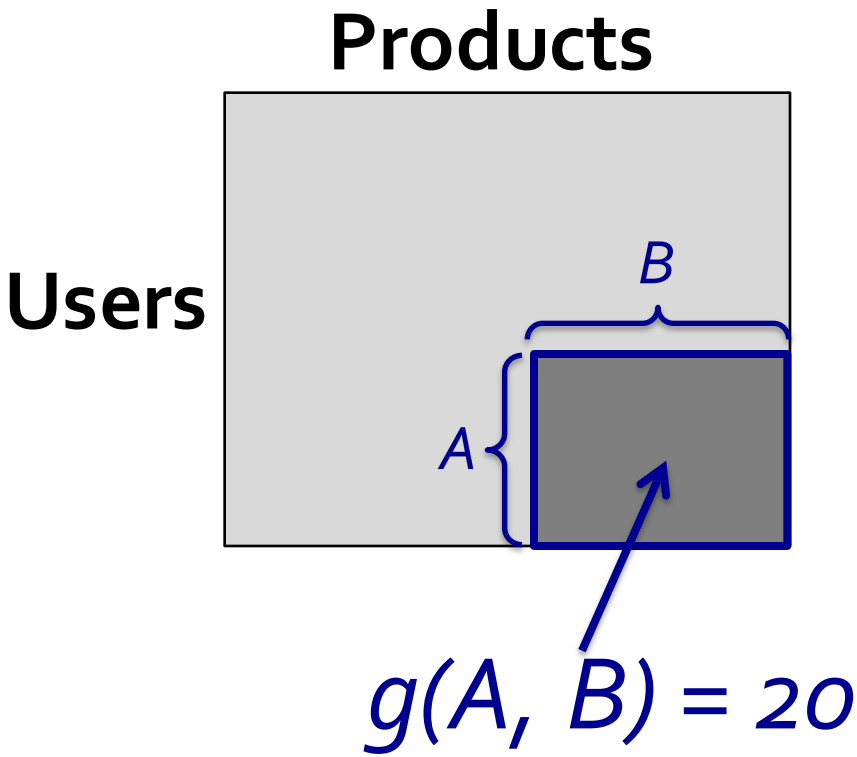
Formal Problem Definition

- Given:
  - Bipartite graph between users and products
  - (optional: prior node suspiciousness scores)
- Detect attacks in a way that is:
  - **Camouflage-resistant**
  - **Near-linear time**
  - **Offers provable bounds**
  - **Works well in practice**



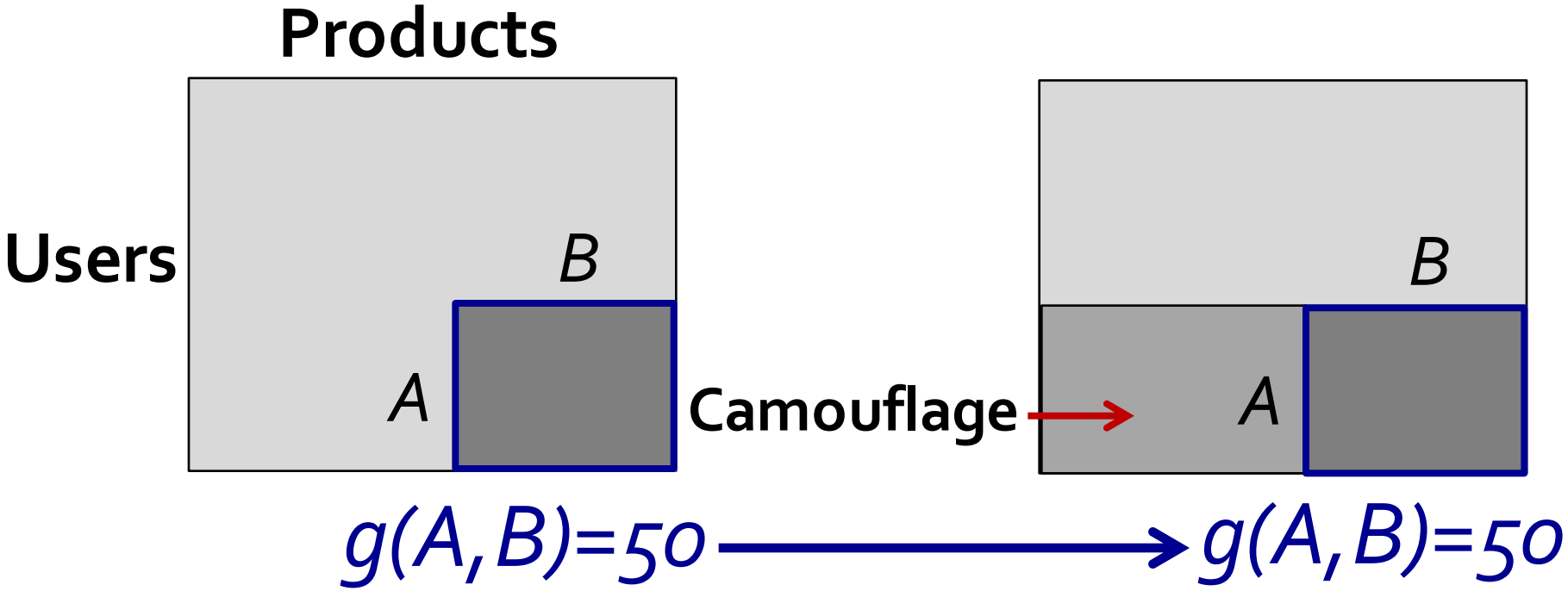
Suspiciousness Metric





Camouflage-Resistance

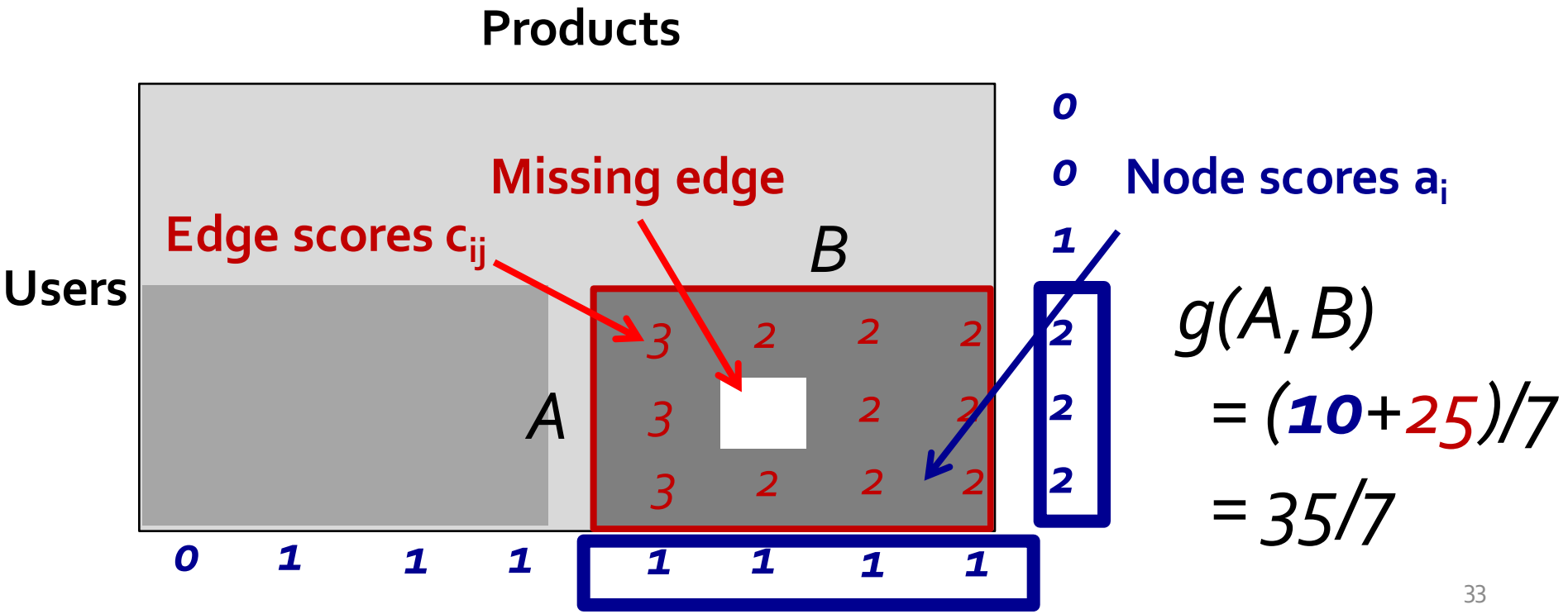
- g is ***camouflage-resistant*** if *g(A,B)* does not decrease when camouflage is added to *A*.



Proposed Suspiciousness Metric

"*Average suspiciousness*" $g(A,B)$

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

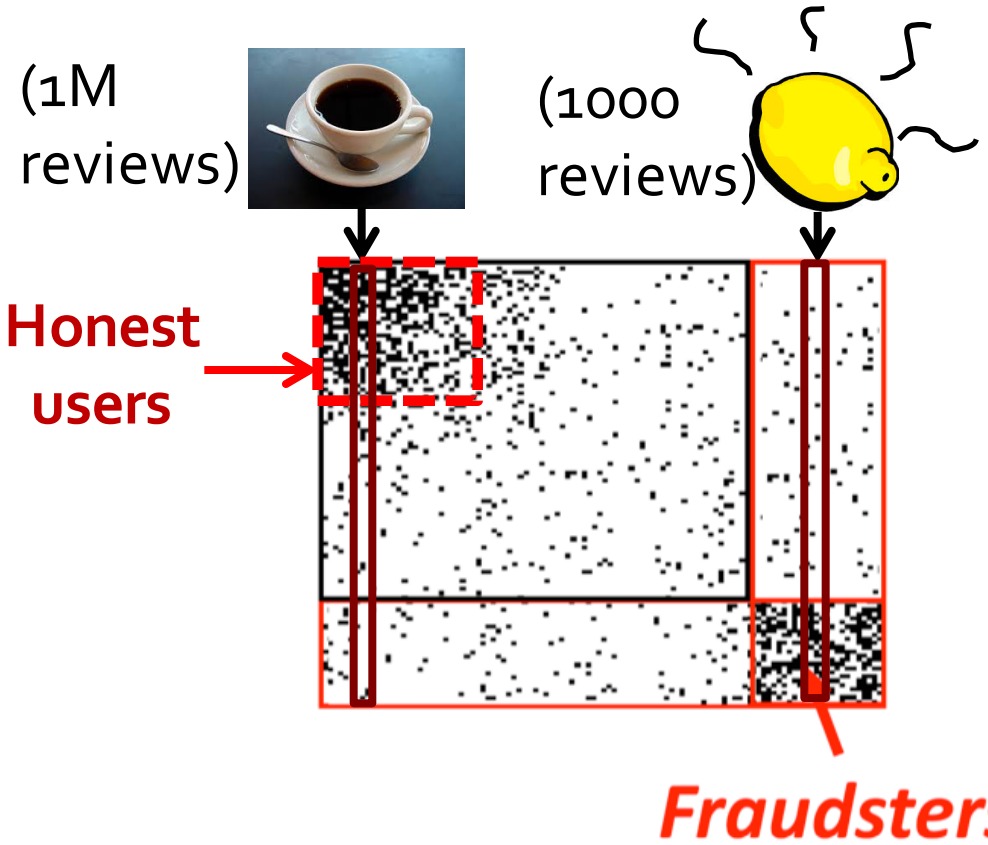


Edge Scores c_{ij}

tf-**idf**

Proposed weighting scheme:

$c_{ij} = 1 / log(\text{unweighted sum of } j\text{-th column})$



Why?

- Popular products are not necessarily suspicious
- Fraudulent products have a high fraction of edges from fraud

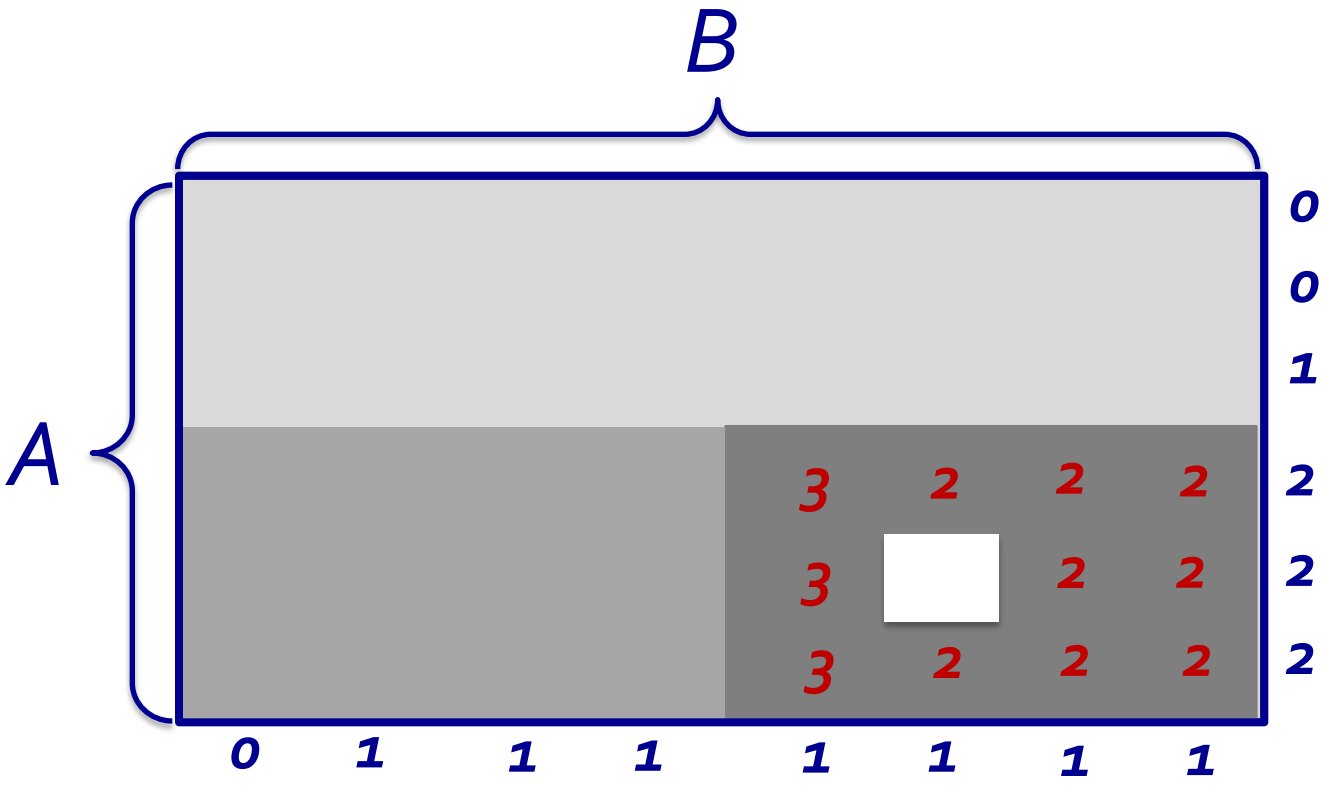
Metric Properties

Average suspiciousness g(A,B):

- ▢ **Can be optimized in near-linear time**
- ▢ Provable bounds
- ▢ Camouflage-resistant
- ▢ Works in practice

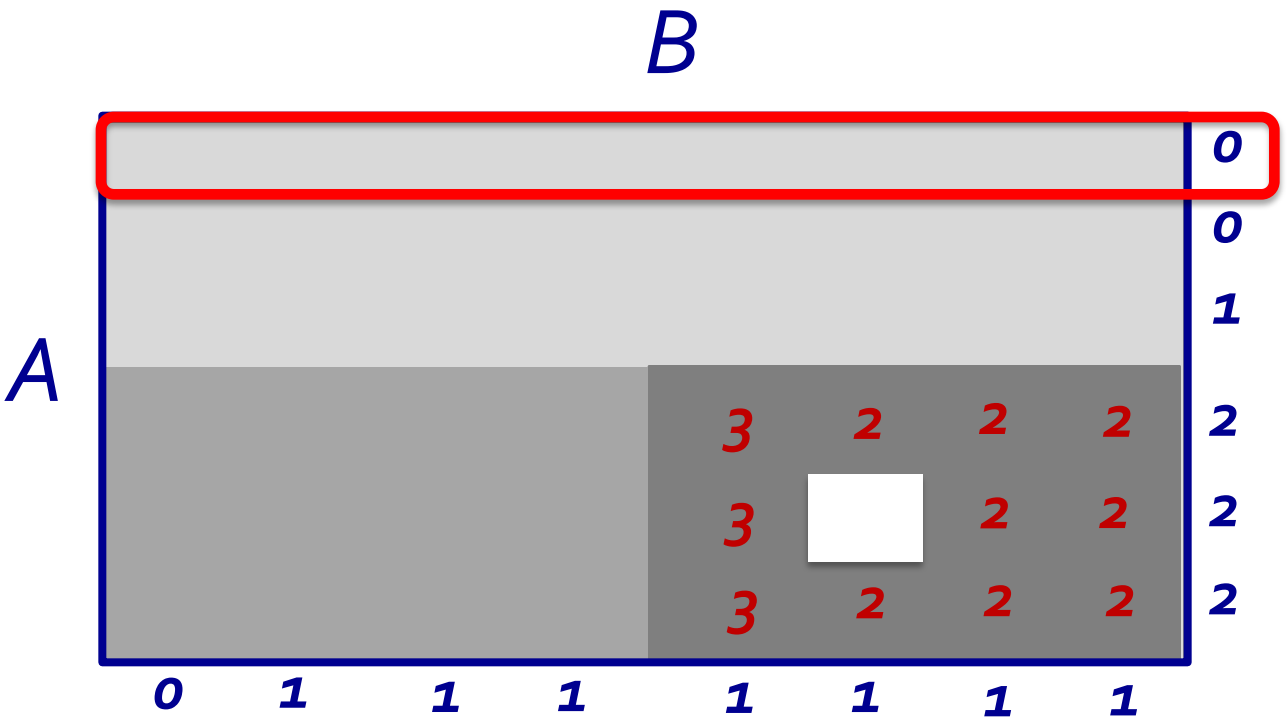
FRAUDAR: Greedy Algorithm

- Start with sets A, B as all users / products



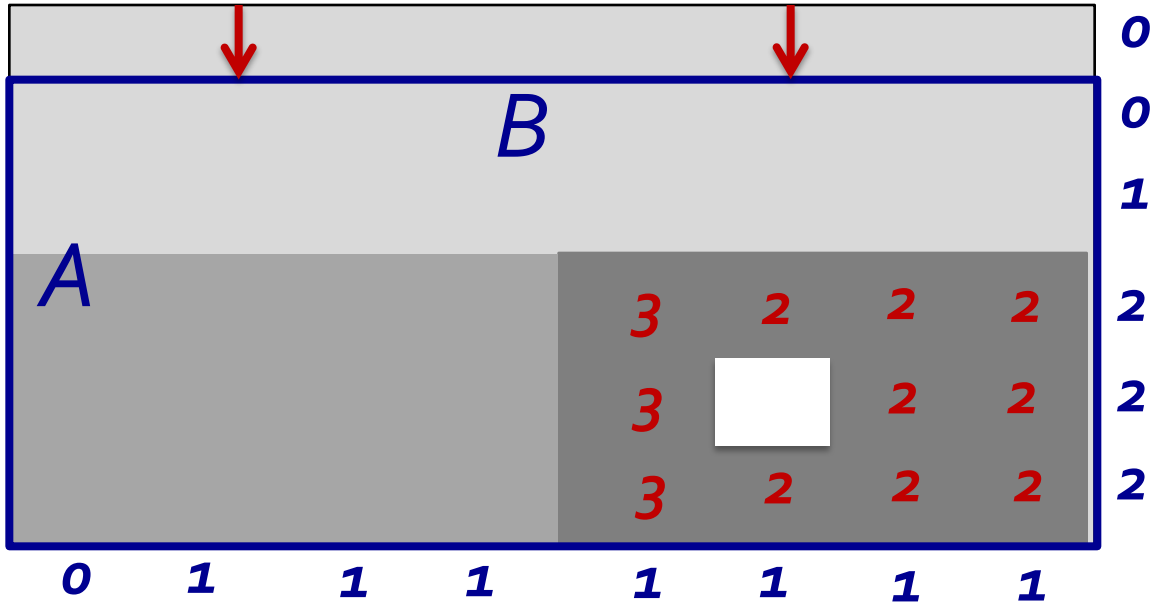
FRAUDAR: Greedy Algorithm (cont.)

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



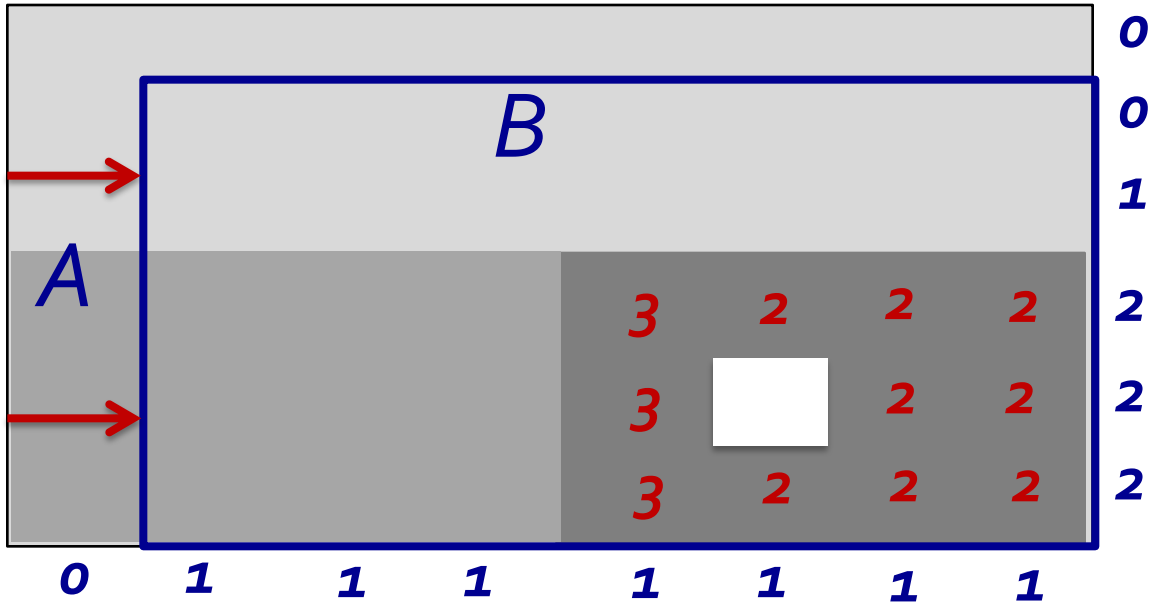
FRAUDAR: Greedy Algorithm (cont.)

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



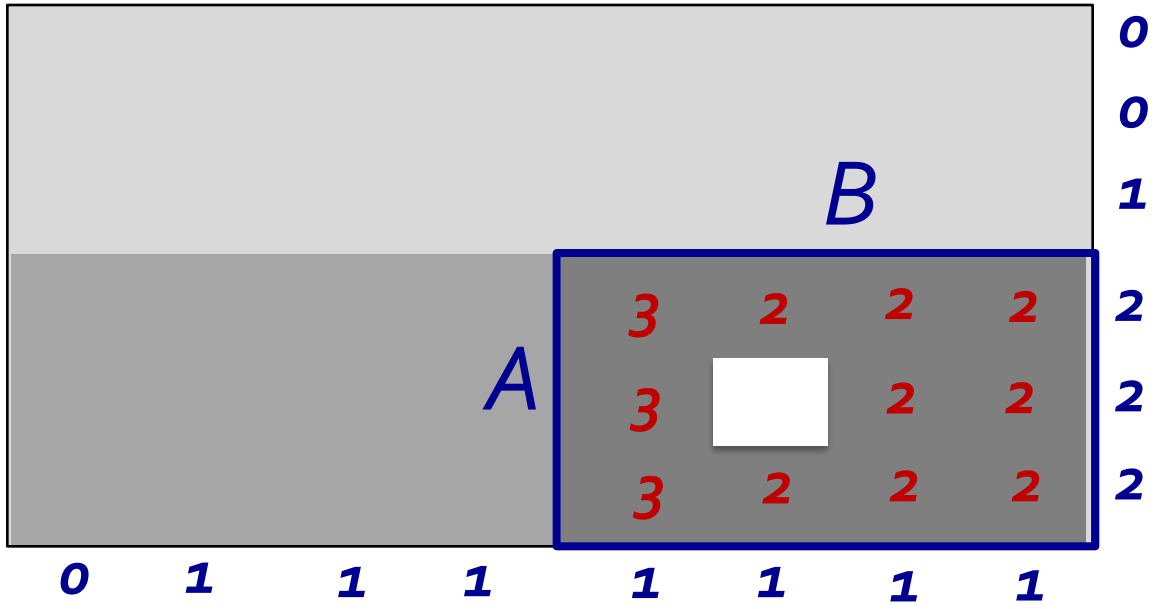
FRAUDAR: Greedy Algorithm (cont.)

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



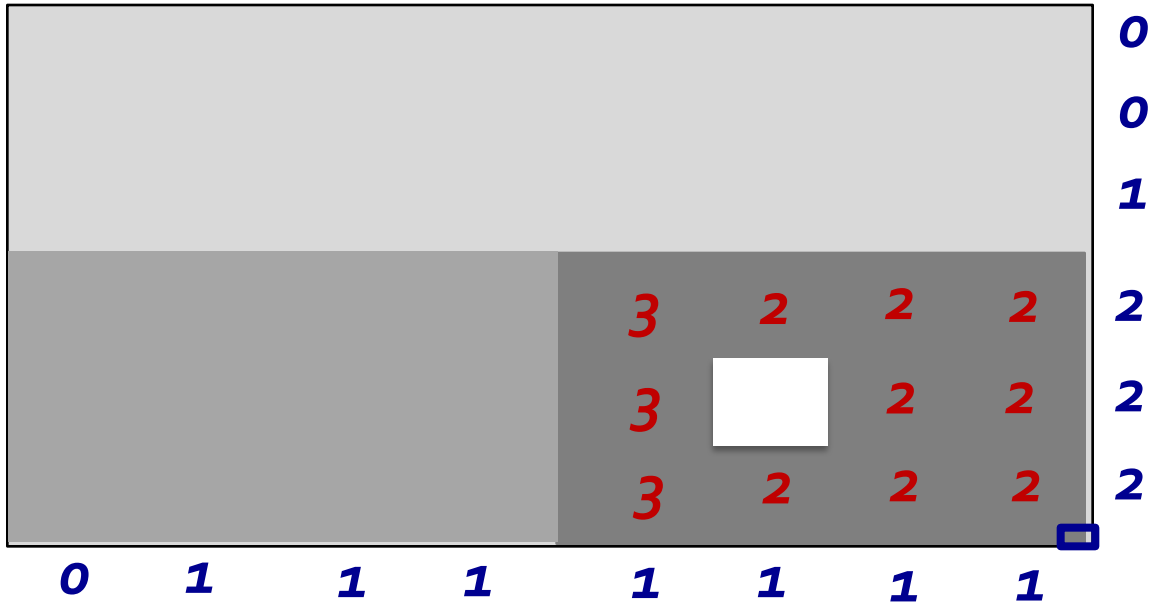
FRAUDAR: Greedy Algorithm (cont.)

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



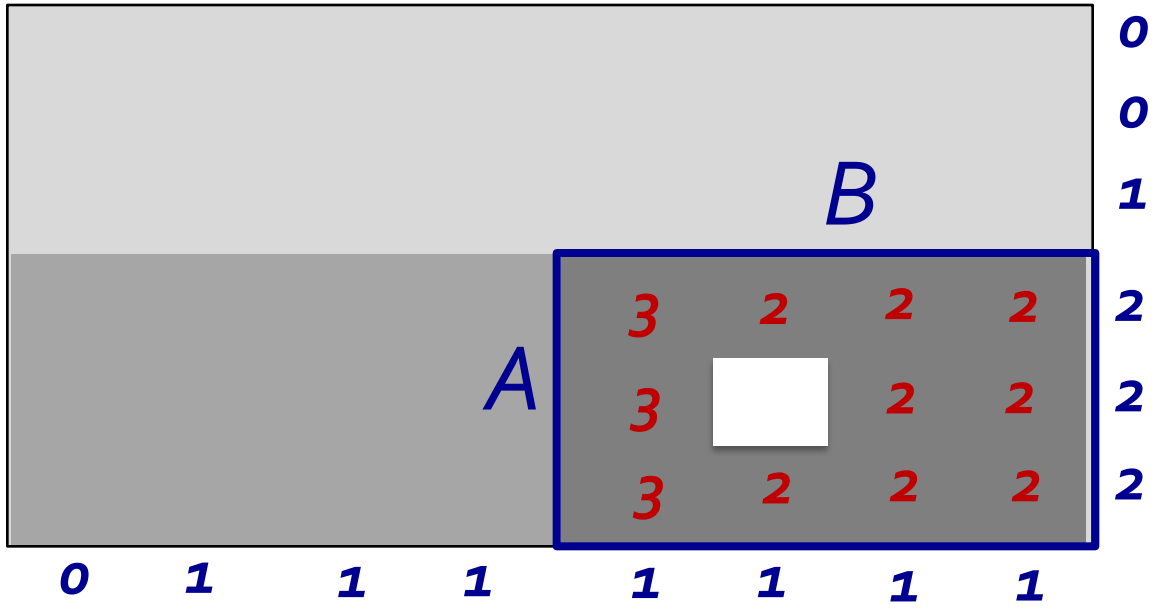
FRAUDAR: Greedy Algorithm (cont.)

- Continue until A and B are empty



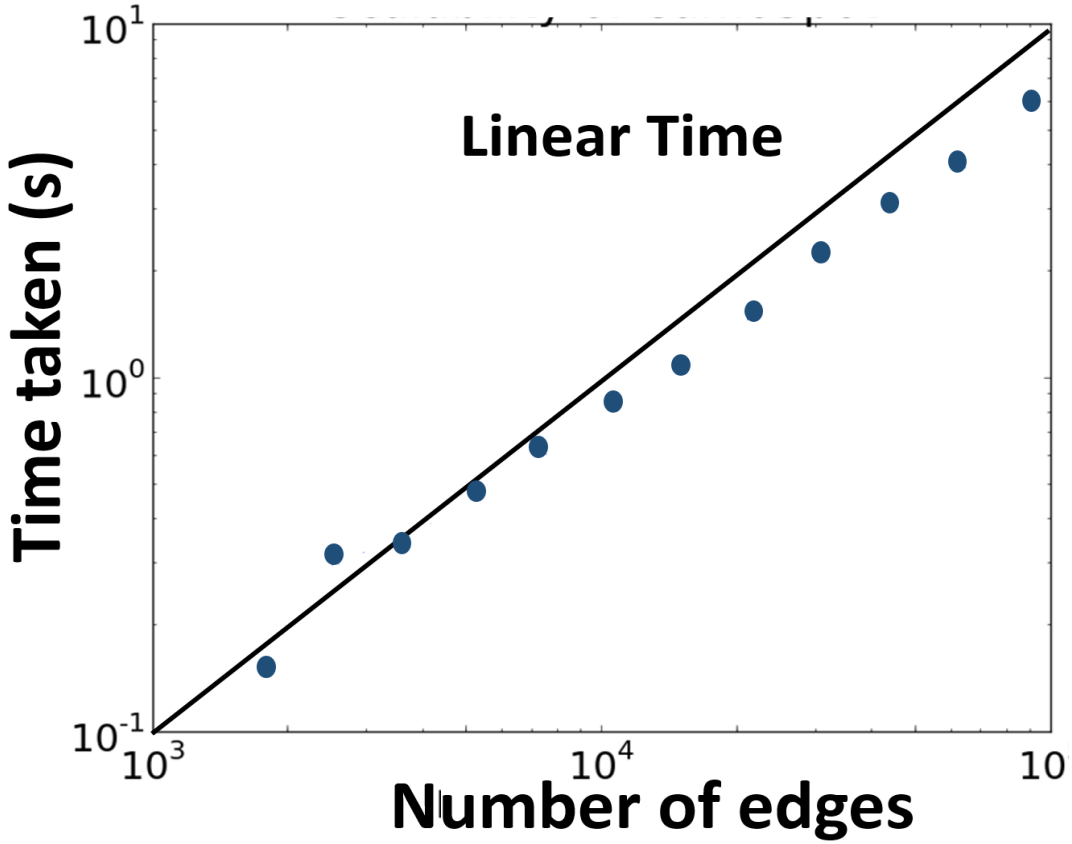
FRAUDAR: Greedy Algorithm (cont.)

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



Computation Time

- O(|E| log(|V|): using appropriate data structures



Metric Properties

Average suspiciousness g(A,B):

- ☑ Can be optimized in near-linear time
- ☐ **Provable bounds**
- ☐ Camouflage-resistant
- ☐ Works in practice

Theoretical Guarantee

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

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

FRAUDAR subgraph

Optimum value of g

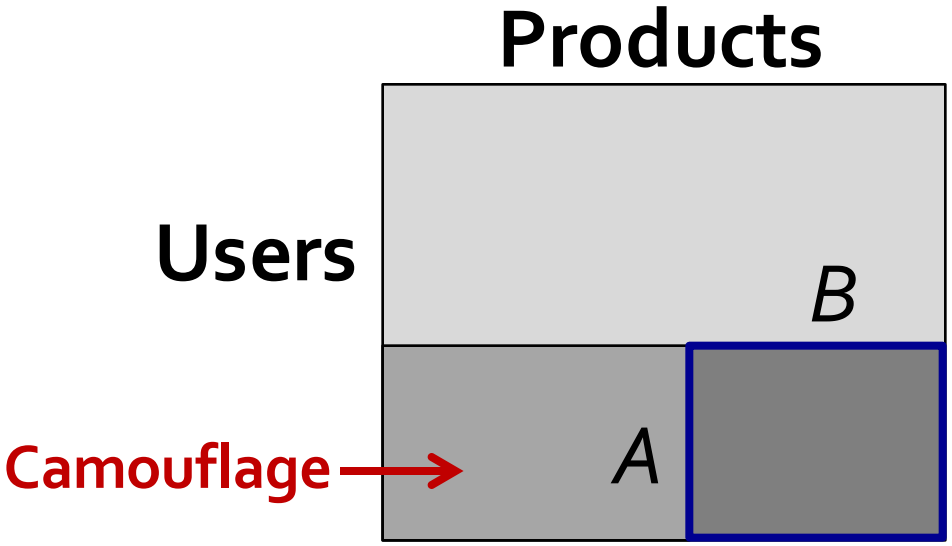
Metric Properties

Average suspiciousness g(A,B):

- ☑ Can be optimized in near-linear time
- ☑ Provable bounds
- ☐ **Camouflage-resistant**
- ☐ Works in practice

Camouflage Resistance

- Theorem 2: If c_{ij} is a *column weighting* (i.e. c_{ij} is any function of the j-th column), then **g is camouflage-resistant**.



(c_{ij} = 1 / log(sum of j-th column) satisfies this)

Metric Properties

Average suspiciousness g(A,B):

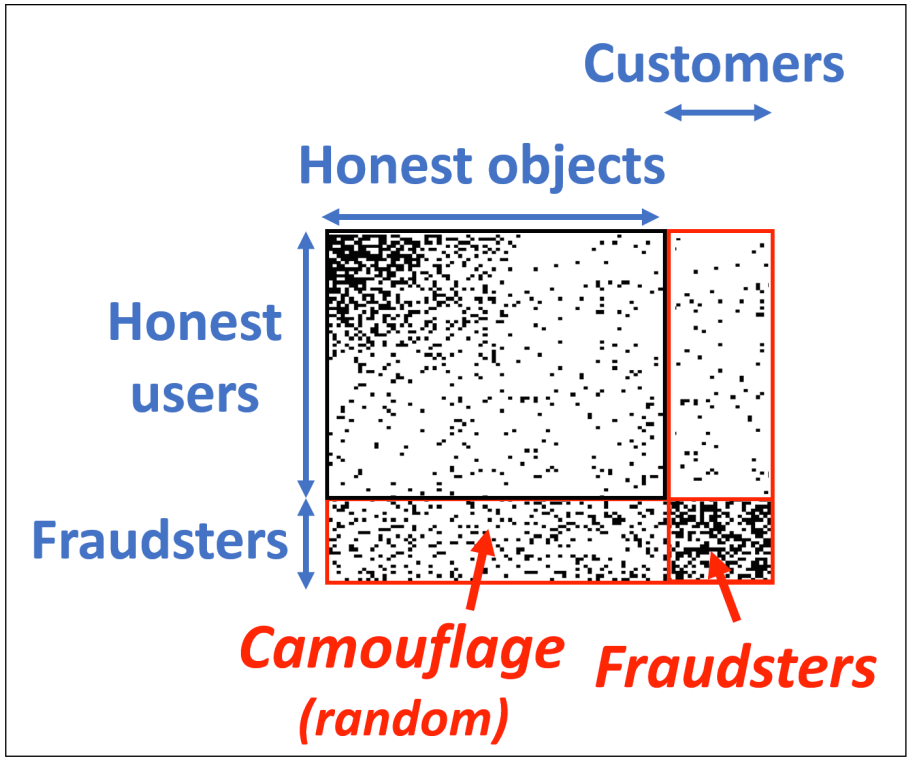
- ☑ Can be optimized in near-linear time
- ☑ Provable bounds
- ☑ Camouflage-resistant
- ☐ **Works in practice**

Experiments: Detecting Injection of Various Types of "Camouflage"

- Amazon Review Graph:
  24K users, 4K products
- Injected 200 x 200 blocks with various types of camouflage
  - None
  - Random camouflage
  - Biased camouflage
  - Hijacked accounts

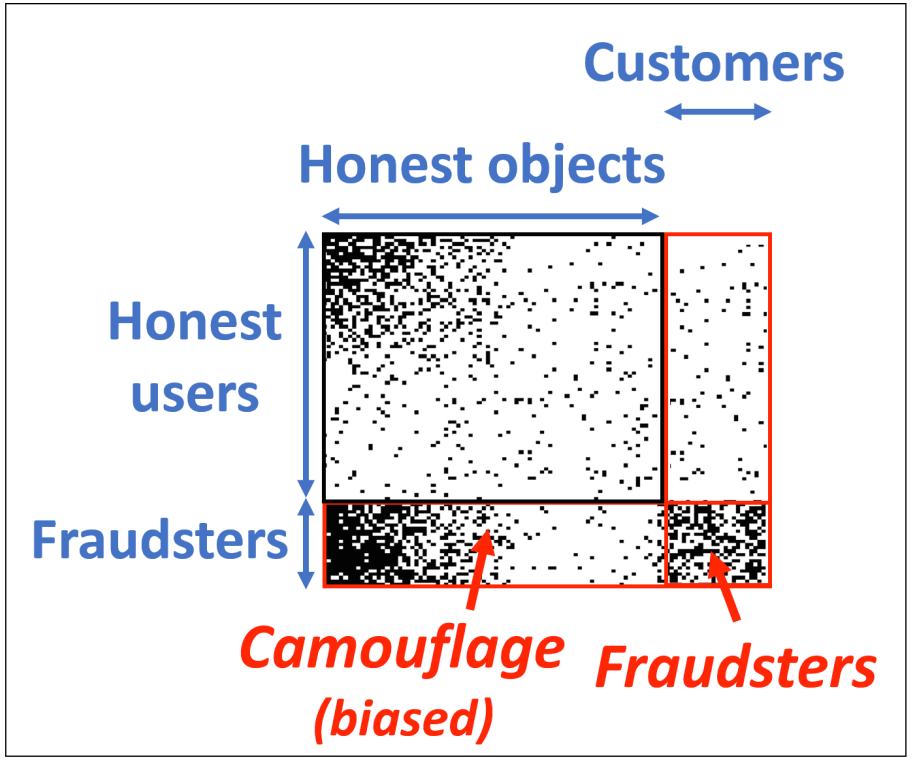
Experiments: Detecting Injection of Various Types of "Camouflage"

- Amazon Review Graph: 24K users, 4K products
- Injected 200 x 200 blocks with various types of camouflage
  - None
  - **Random camouflage**
  - Biased camouflage
  - Hijacked accounts



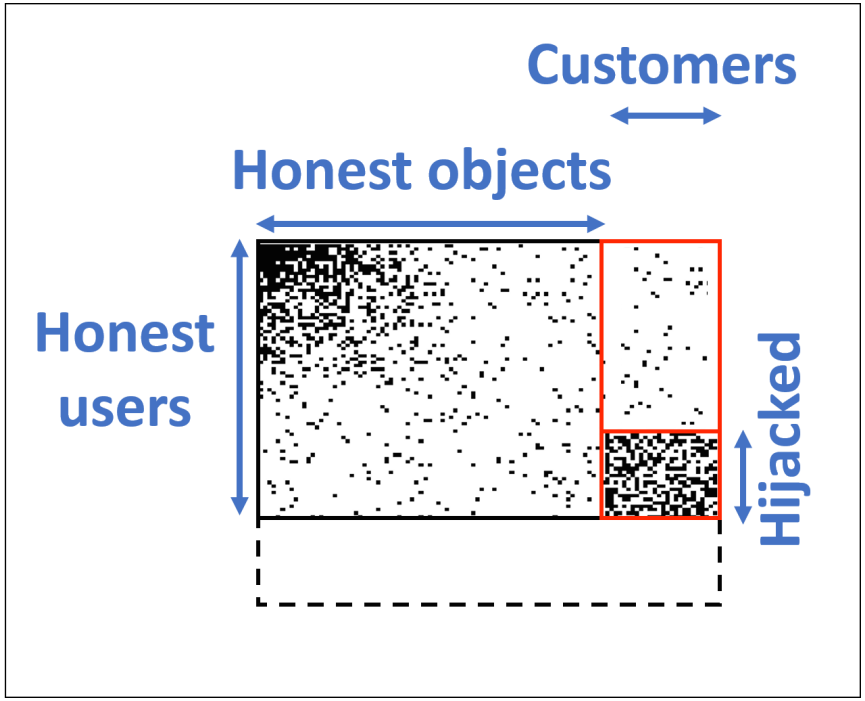
Experiments: Detecting Injection of Various Types of "Camouflage"

- Amazon Review Graph: 24K users, 4K products
- Injected 200 x 200 blocks with various types of camouflage
  - None
  - Random camouflage
  - **Biased camouflage**
  - Hijacked accounts

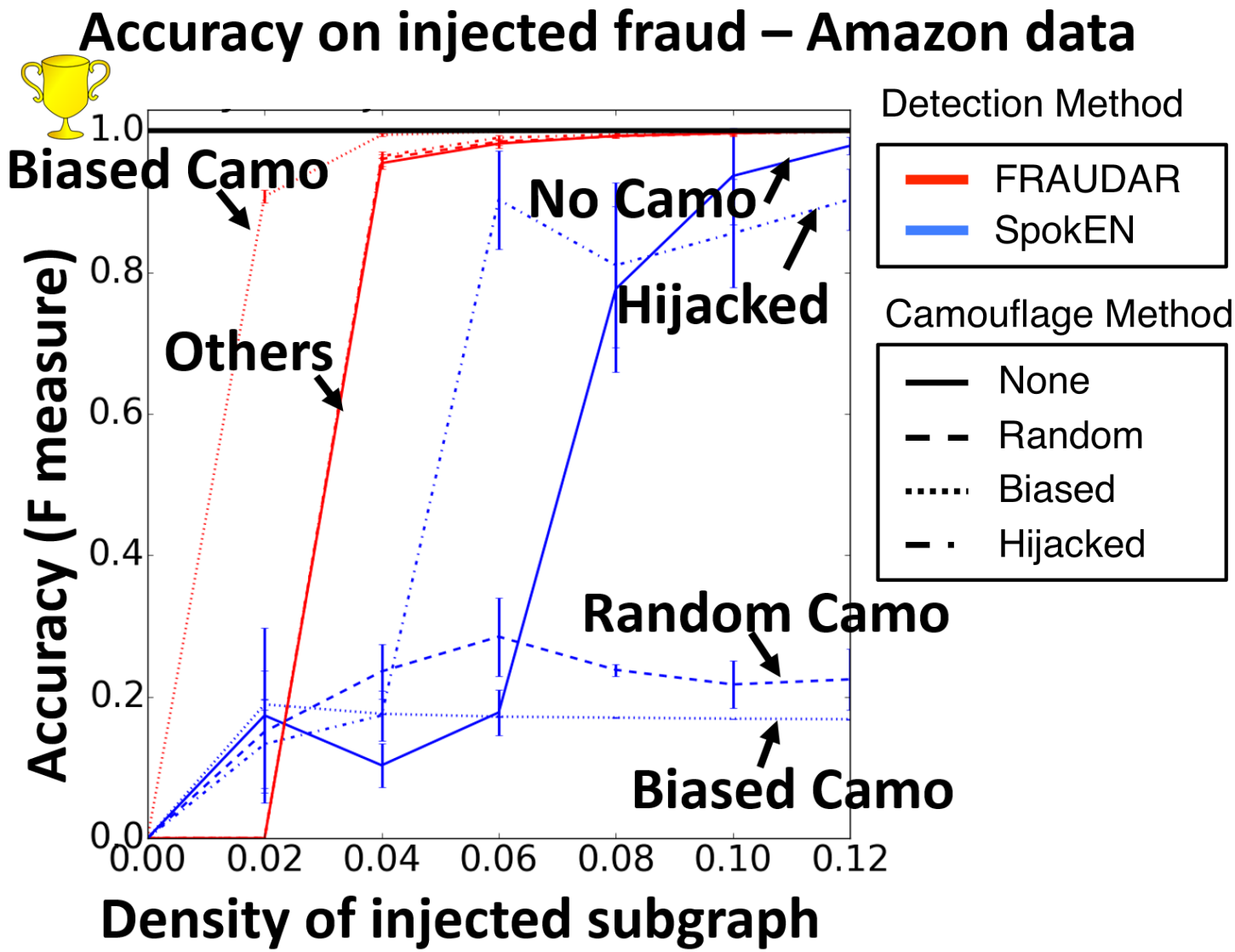


Experiments: Detecting Injection of Various Types of "Camouflage"

- Amazon Review Graph: 24K users, 4K products
- Injected 200 x 200 blocks with various types of camouflage
  - None
  - Random camouflage
  - Biased camouflage
  - **Hijacked accounts**



Accuracy on Detecting Injected Fraud



Summary

- Spectral methods
  - Spectral clustering and community detection
- Spectral subspaces and spectral subspace plots
- EigenSpokes (singular vectors and "spokes")
- LockInfer ("**camouflage**", "fame", "pearls", "staircase", etc.)
- fBox (small-scale, stealthy attacks; reconstructed degrees)
- FRAUDAR (theoretical guarantees for bounding graph fraud in the face of **camouflage**)
- Applications: Mobile calls, Twitter social network, "user-product" reviews

References

- Andrew Y. Ng, Michael I. Jordan, Yair Weiss. "On Spectral Clustering: Analysis and an algorithm", NIPS 2001.
- Scott White and Padhraic Smyth. "A spectral clustering approach to finding communities in graphs", SDM 2005.
- M. E. J. Newman. "Finding community structure in networks using the eigenvectors of matrices", Physical Review E 2006.
- Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju, Christos Faloutsos. "EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs", PAKDD 2010.
- Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang. "Inferring lockstep behavior from connectivity pattern in large graphs", KAIS 2016.
- Neil Shah, Alex Beutel, Brian Gallagher, Christos Faloutsos. "Spotting suspicious link behavior with fBox: An adversarial perspective", ICDM 2014.
- Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, Christos Faloutsos. "FRAUDAR: Bounding Graph Fraud in the Face of Camouflage", KDD 2016 Best Research Paper Award.

KDD 2017
Halifax, Nova Scotia - Canada
August 13 - 17, 2017

Tutorial: Data-Driven Approaches towards Malicious Behavior Modeling

Meng Jiang
University of Notre Dame

Srijan Kumar
Stanford University

Christos Faloutsos
Carnegie Mellon University

V.S. Subrahmanian
University of Maryland, College Park

http://www.meng-jiang.com/tutorial-kdd17.html