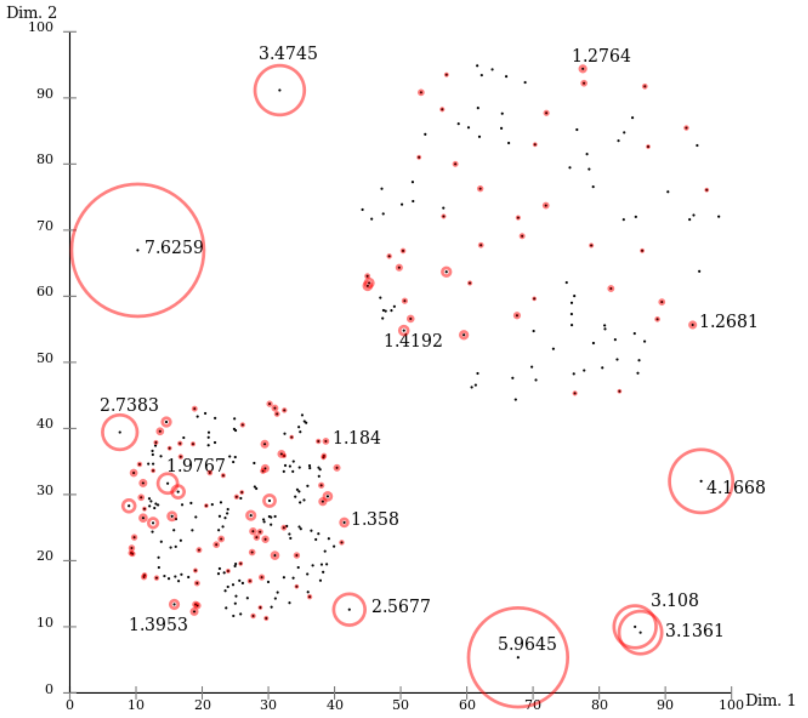
What is Strange in Large Networks? Graph-based Irregularity and Fraud Detection

Leman Akoglu

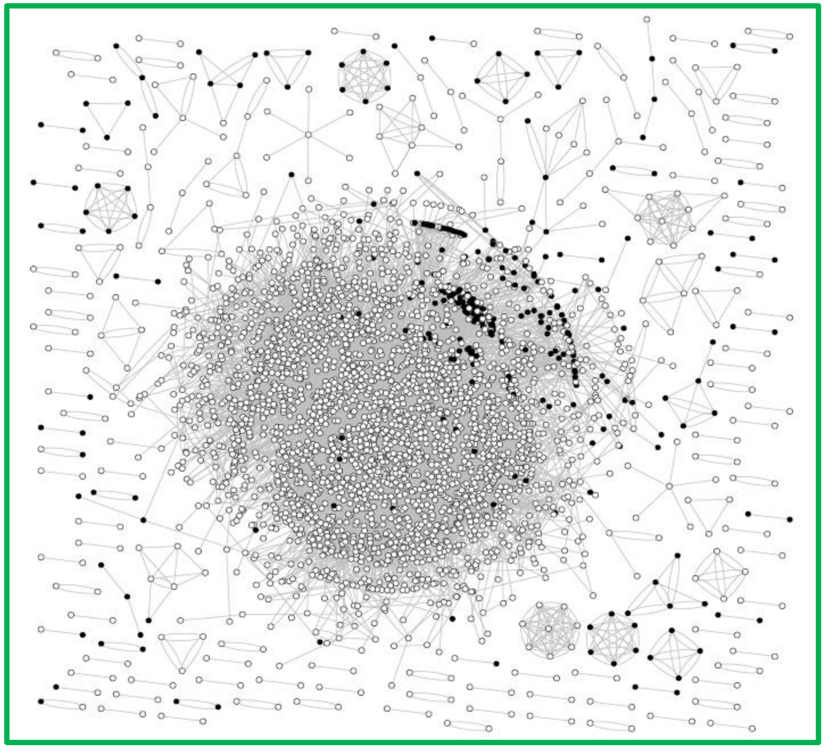
Christos Faloutsos

Outliers vs. Graph anomalies



Clouds of points
(multi-dimensional)

This tutorial



Inter-linked objects
(network)

Roadmap

9:00	**Part I.I:** Anomaly detection in static data
10:30	**Coffee break**
11:00	**Part I.II:**
13:00	**Lunch**
14:30	**Part II:** Anomaly detection in dynamic data
16:00	**Coffee break**
16:30	**Part III:** Graph-based algorithms & applications
17:30	The End

Disclaimers

References are not necessarily authoritative and complete

Let me know of additional related work at **leman@cs.stonybrook.edu**

Several slides have been reused or modified by the permission of the original creators.

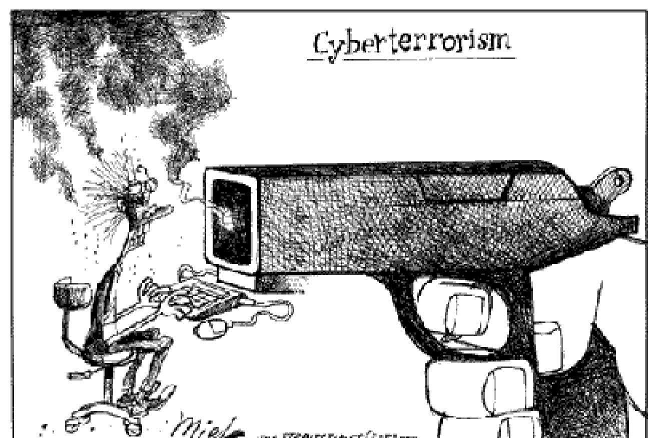
Anomaly detection: Applications

Tax evasion

Credit card fraud

Healthcare fraud

Network intrusion



Applications

Malware

Investment fraud

Click fraud

Spyware

Insurance fraud

Malicious cargo

Auction fraud

Damage detection

Fake reviews

Medical diagnosis

Email spam

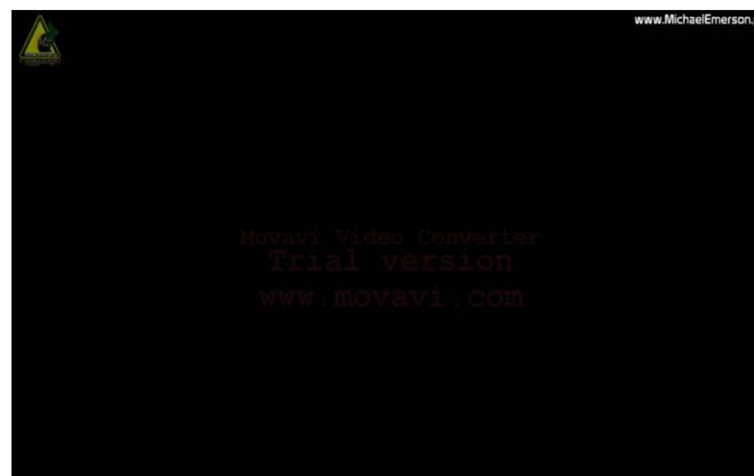
False advertising

Performance monitoring

Web spam

Insider threat

Image/video surveillance



Anomaly detection: definition

- (Hawkins' Definition of Outlier, 1980)

  "An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism."

No unique definition → leads to → Many definitions in various contexts

outlier, anomaly, outbreak, event, fraud, …

Anomaly detection: definition

- for practical purposes,

  a record/point/graph-node/graph-edge

  is flagged as anomalous

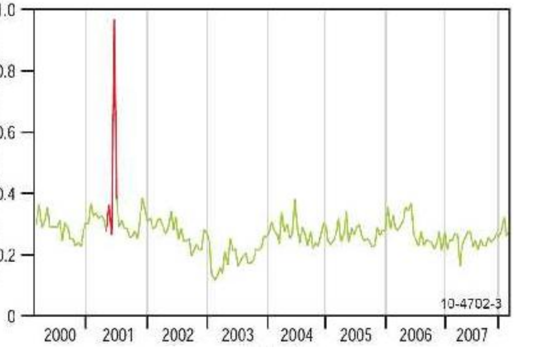
  if a rarity/likelihood/outlierness score

  exceeds a user-defined threshold

- anomalies:

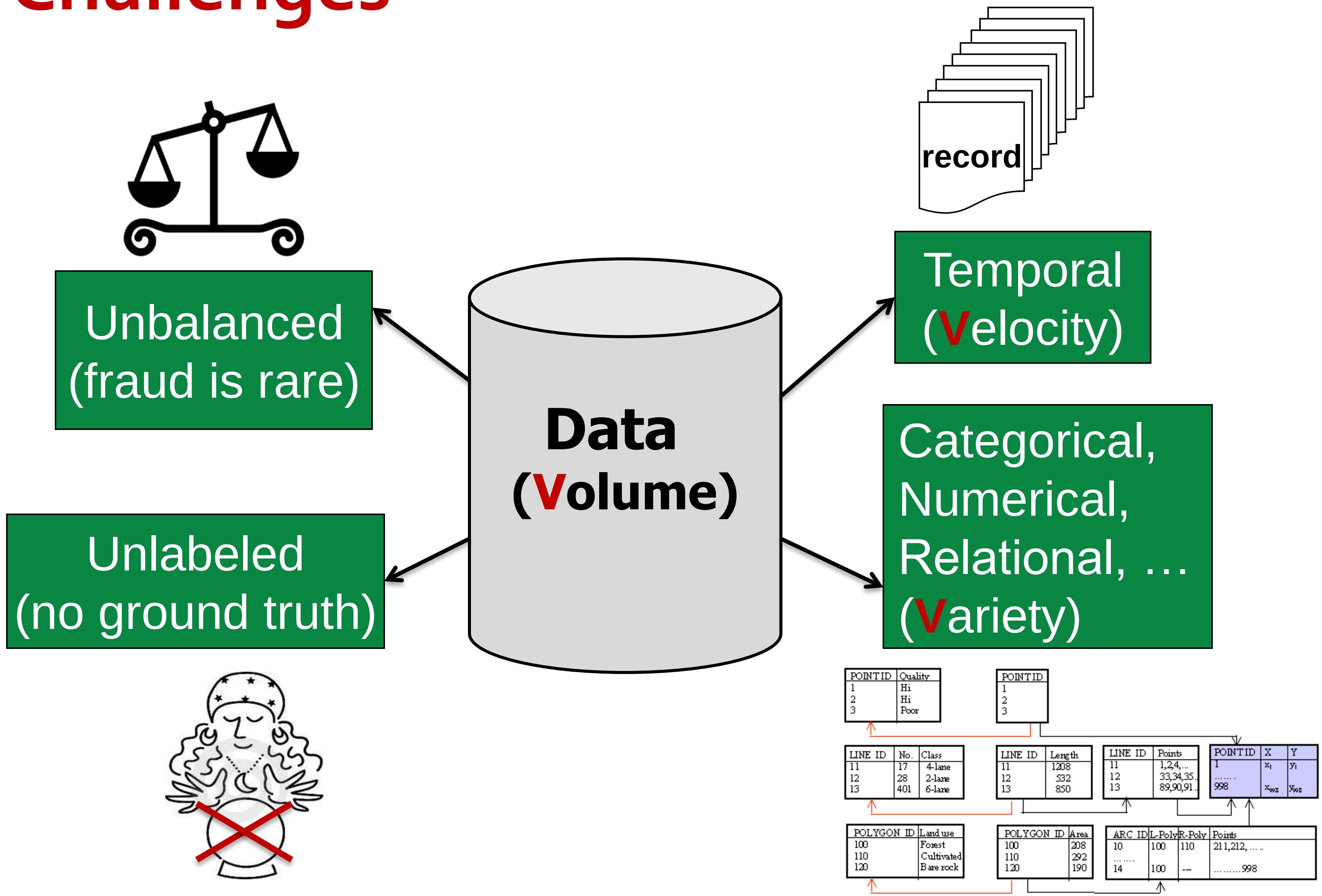
  → rare (e.g., rare combination of categorical attribute values)

  → isolated points in n-d spaces



  → surprising (don't fit well in our mental/statistical model == need too many bits under MDL)

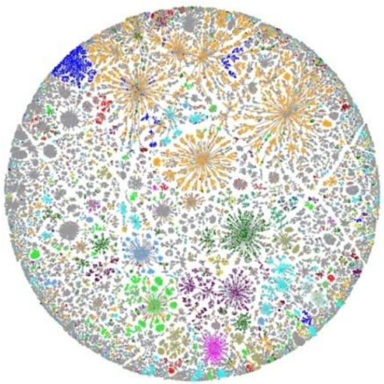
Challenges


record
Temporal
(Velocity)
Unbalanced
(fraud is rare)
Data
(Volume)
Categorical,
Numerical,
Relational, …
(Variety)
Unlabeled
(no ground truth)

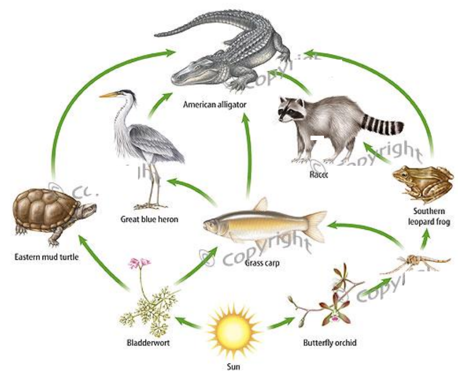
Why graph-based detection?

- Powerful representation
  - Interdependent instances
  - Long-range relations
  - Node/Edge attributes (data complexity)
  - Hard to fake/alter (adversarial robustness)
- Abundant relational data
  - Web, email, phone call, …
- Nature of applications
  - (1) opportunistic fraud (word of mouth)
  - (2) organized fraud (group activity)

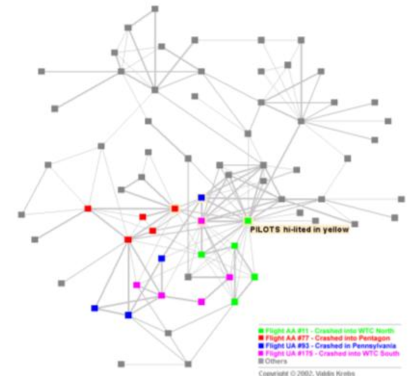
Real graphs (1)



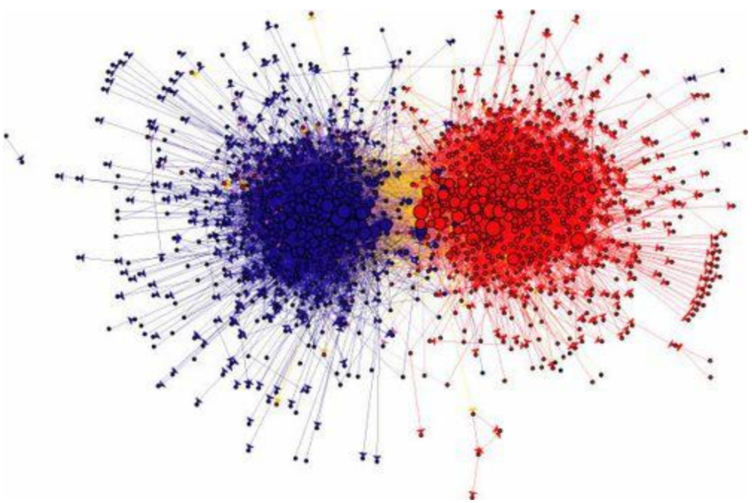
Internet Map



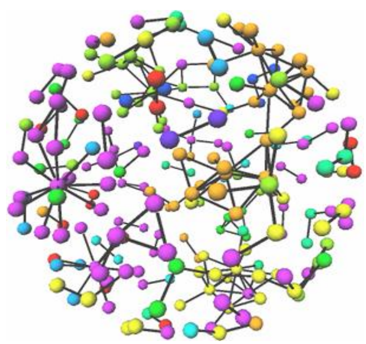
Food Web



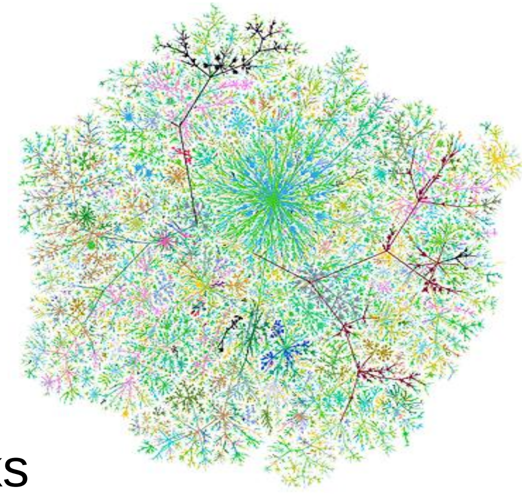
Terrorist Network



Blog networks

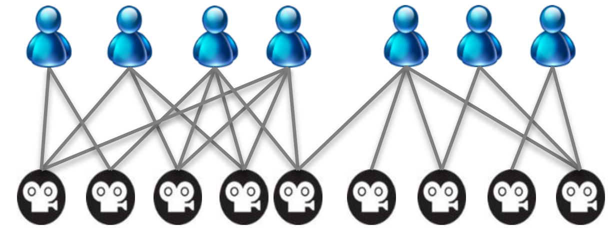


Biological networks

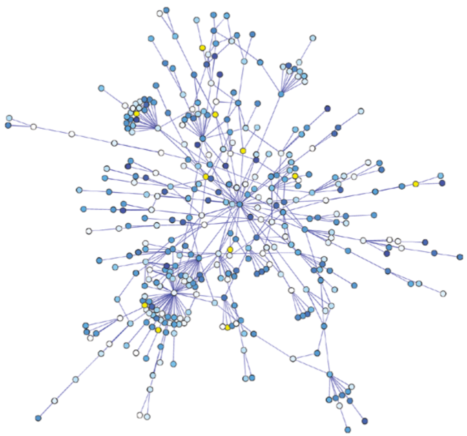


Web Graph

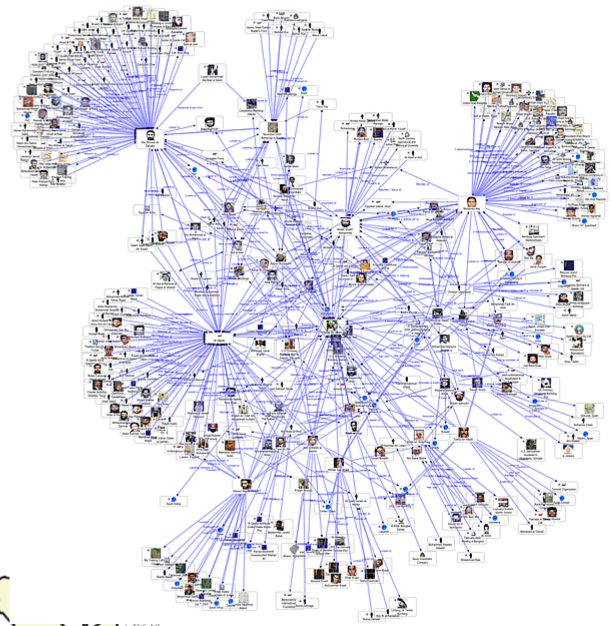
Real graphs (2)



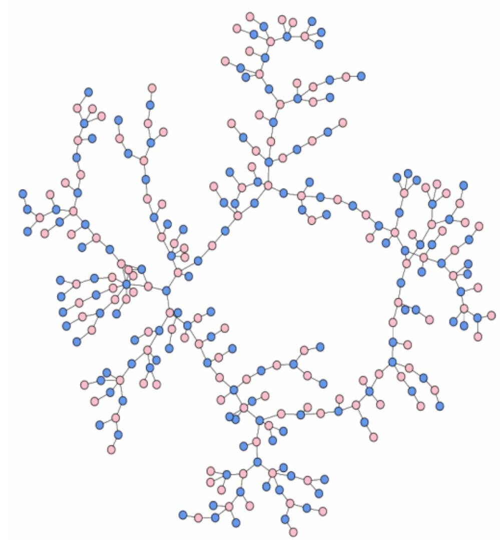
Retail networks



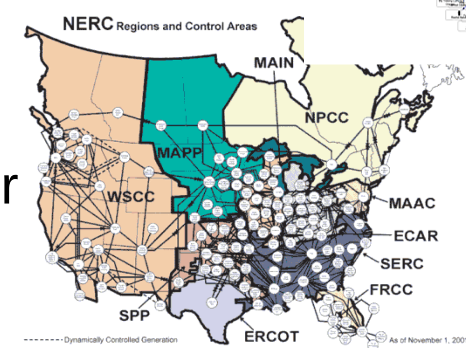
Protein-protein
Interaction



Social Network



Dating network

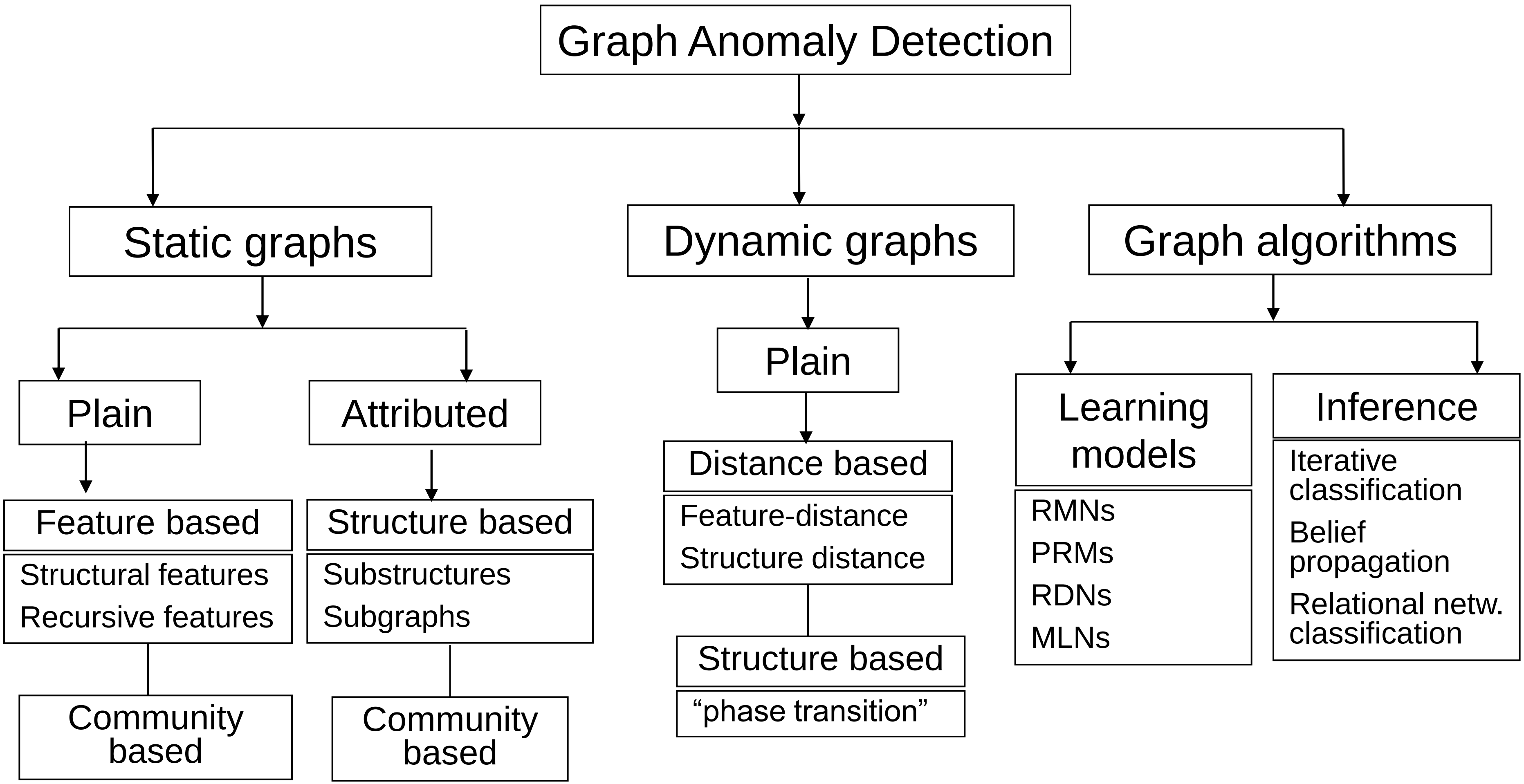


Power
Grid

Problem revisited for graphs

- Three different problem settings
  - Unlabeled/Labeled (Attributed) Graphs
  - Static/Dynamic Graphs
  - Un-/Semi-/- Supervised Graph Techniques

Taxonomy



Graph Anomaly Detection
Static graphs
Dynamic graphs
Graph algorithms
Plain
Attributed
Feature based
Structural features
Recursive features
Community based
Structure based
Substructures
Subgraphs
Community based
Plain
Distance based
Feature-distance
Structure distance
Structure based
“phase transition”
Learning models
RMNs
PRMs
RDNs
MLNs
Inference
Iterative classification
Belief propagation
Relational netw. classification

Goal of this tutorial

- Introduce various problem formulations
  - Definitions change by application/representation
- Applications of problem settings
  - Intrusion, fraud, spam
- Introduce existing techniques
  - Model fitting, factorization, relational inference
- Pros and Cons
  - Parameters, scalability, robustness

Tutorial Outline

- Motivation, applications, challenges
- **Part I:** Anomaly detection in **static** data
  - Overview: Outliers in **clouds of points**
  - Anomaly detection in **graph data**
- **Part II:** Event detection in **dynamic** data
  - Overview: Change detection in **time series**
  - Event detection in **graph sequences**
- **Part III:** Graph-based **algorithms and apps**
  - Algorithms: **relational learning**
  - Applications: **fraud and spam** detection

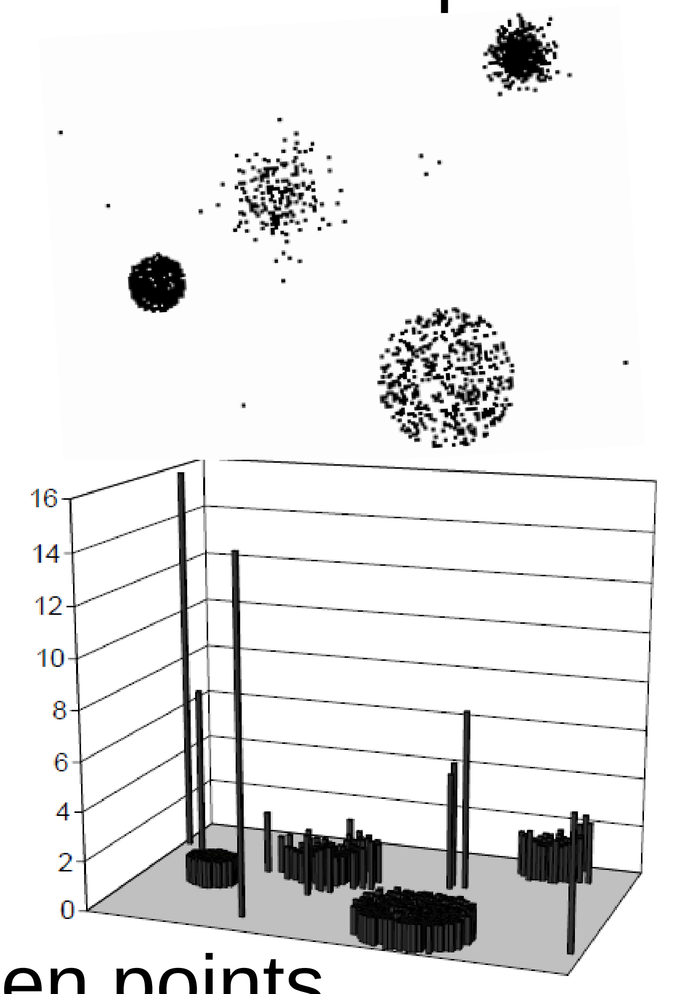
Part I: Anomaly detection in static graphs

Part I: Outline

- Overview: Outliers in **clouds of points**
  - Outliers in **numerical** data points
    - distance-based, density-based, …
  - Outliers in **categorical** data points
    - model-based
- Anomaly detection in **graph data**
  - Anomalies in unlabeled, **plain** graphs
  - Anomalies in node-/edge-labeled, **attributed** graphs

Outlier detection

- Anomalies in multi-dimensional data points
  - Density-based
  - Distance-based
  - Depth-based
  - Distribution-based
  - Clustering-based
  - Classification-based
  - Information theory-based
  - Spectrum-based
  - …
- No relational links between points



Part I: References (outliers)

- M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. SIGMOD, 2000.
- S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. ICDE, 2003.
- C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. SIGMOD, 2001.
- A. Ghoting, S. Parthasarathy and M. Otey, Fast Mining of Distance Based Outliers in High-Dimensional Datasets. DAMI, 2008.
- Y. Wang, S. Parthasarathy and S. Tatikonda, Locality Sensitive Outlier Detection. ICDE, 2011.
- Kaustav Das, Jeff Schneider. Detecting Anomalous Records in Categorical Datasets. KDD 2007.

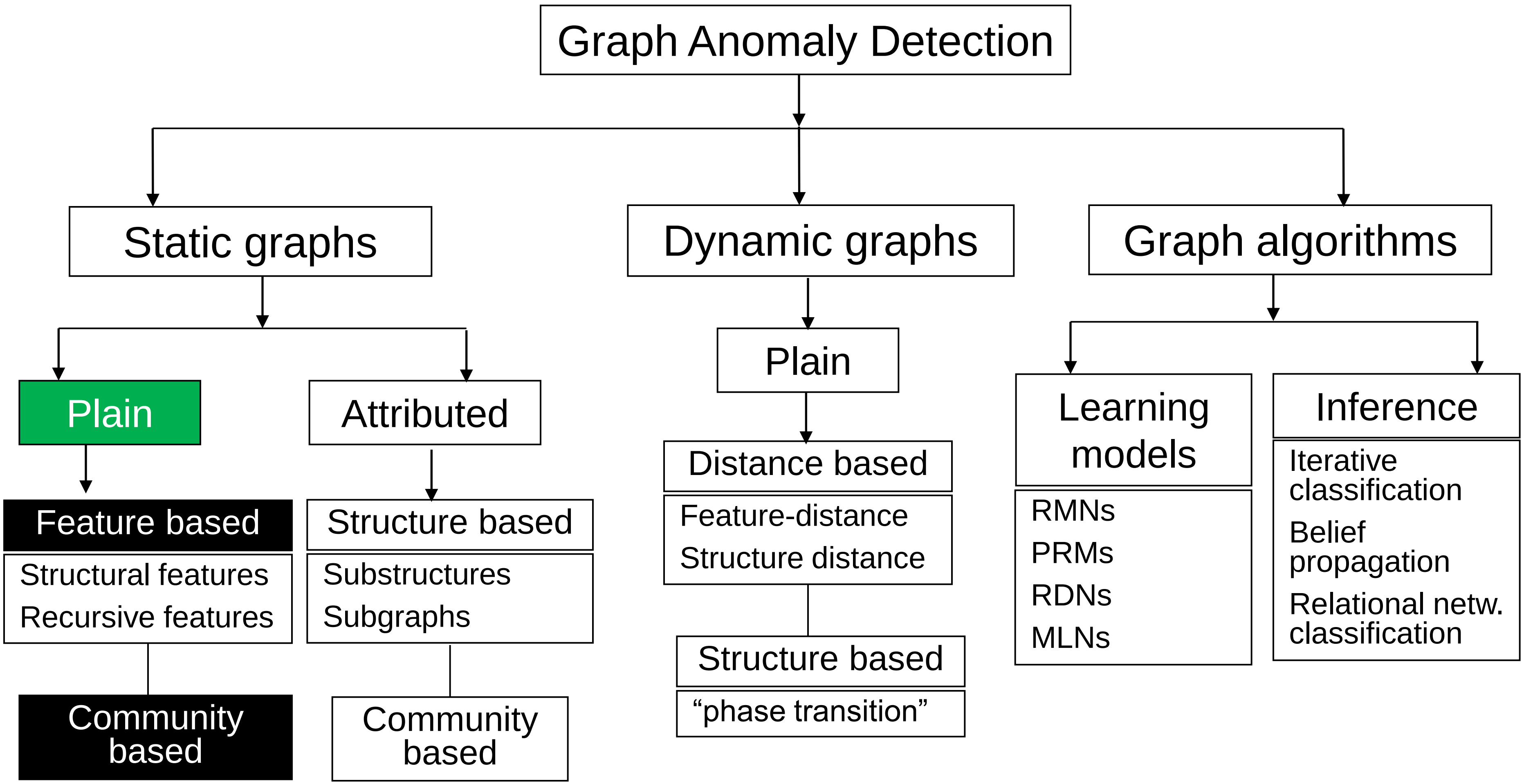
Part I: References (outliers)

- Müller E., Schiffer M., Seidl T. Adaptive Outlierness for Subspace Outlier Ranking. CIKM, 2010.
- Müller E., Assent I., Iglesias P., Mülle Y., Böhm K. Outlier Ranking via Subspace Analysis in Multiple Views of the Data. ICDM, 2012.
- L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos. Fast and Reliable Anomaly Detection in Categoric Data. CIKM, 2012.
- A. Chaudhary, A. S. Szalay, and A. W. Moore. Very fast outlier detection in large multidimensional data sets. DMKD, 2002.
- Survey: V. Chandola, A. Banerjee, V. Kumar: Anomaly Detection: A Survey. ACM Computing Surveys, Vol. 41(3), Article 15, July 2009.

Part I: Outline

- Overview: Outliers in **clouds of points**
  - Outliers in **numerical** data points
    - distance-based, density-based, …
  - Outliers in **categorical** data points
    - model-based
- Anomaly detection in **graph data**
  - Anomalies in unlabeled, **plain** graphs
  - Anomalies in node-/edge-labeled, **attributed** graphs

Taxonomy



Graph Anomaly Detection
Static graphs
Dynamic graphs
Graph algorithms
Plain
Attributed
Feature based
Structural features
Recursive features
Community based
Structure based
Substructures
Subgraphs
Community based
Plain
Distance based
Feature-distance
Structure distance
Structure based
"phase transition"
Learning models
RMNs
PRMs
RDNs
MLNs
Inference
Iterative classification
Belief propagation
Relational netw. classification

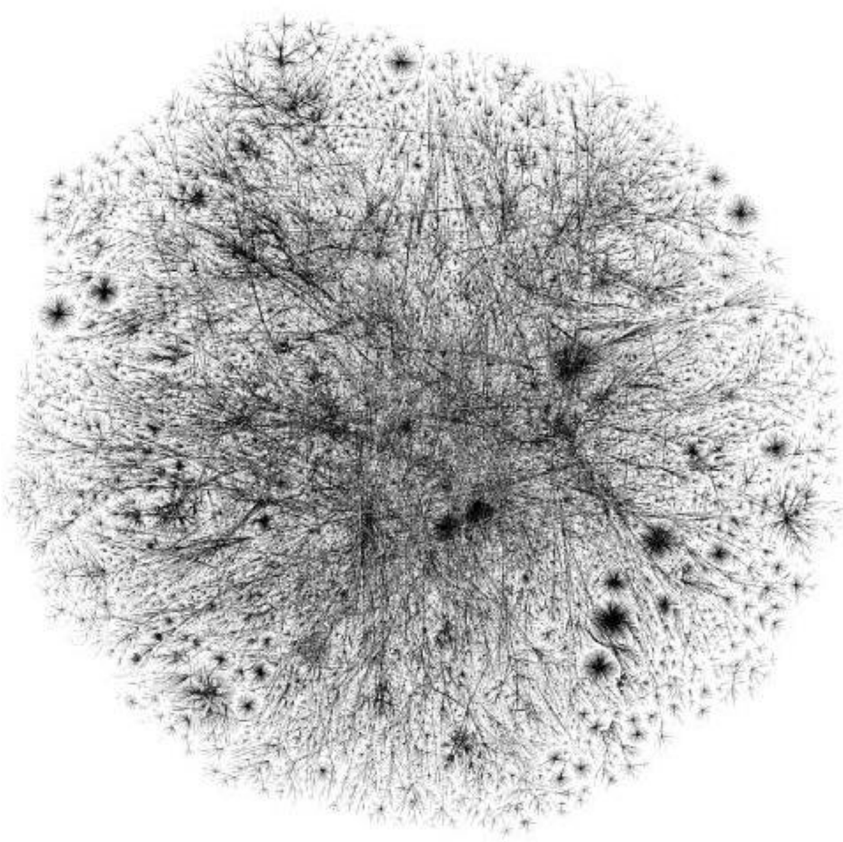
Anomalies in Weighted Graphs

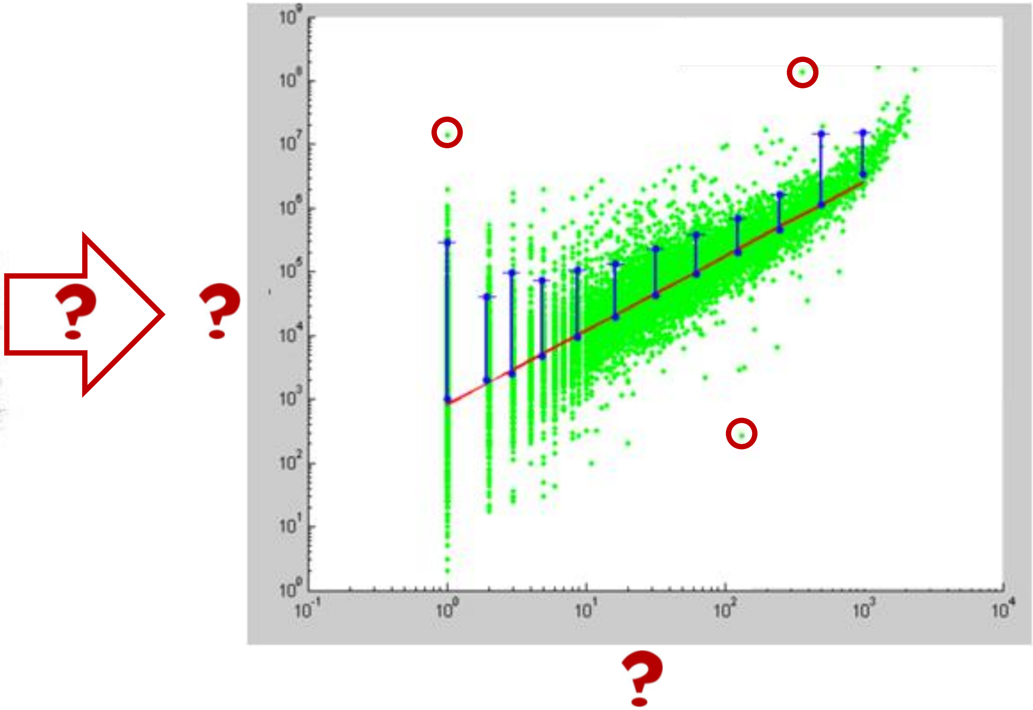
- Problem:

Q1. Given a **weighted** and unlabeled graph, how can we spot strange, abnormal, extreme nodes?

Q2. Can we explain why the spotted nodes are anomalous?

Problem sketch





OddBall: approach

1) For each node,

   1.1) Extract "ego-net" (=1-step neighborhood)

   1.2) Extract features (#edges, total weight, etc.)

      → features that could yield "laws"

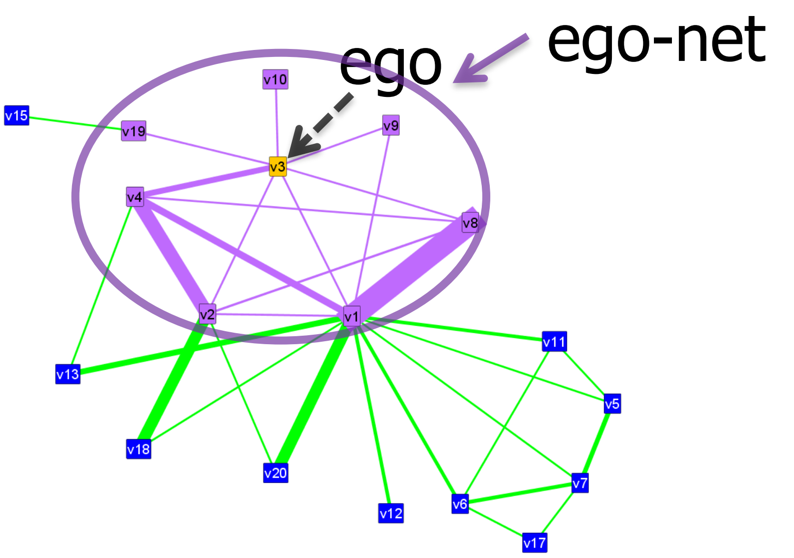
      → features fast to compute and interpret

2) Detect patterns:

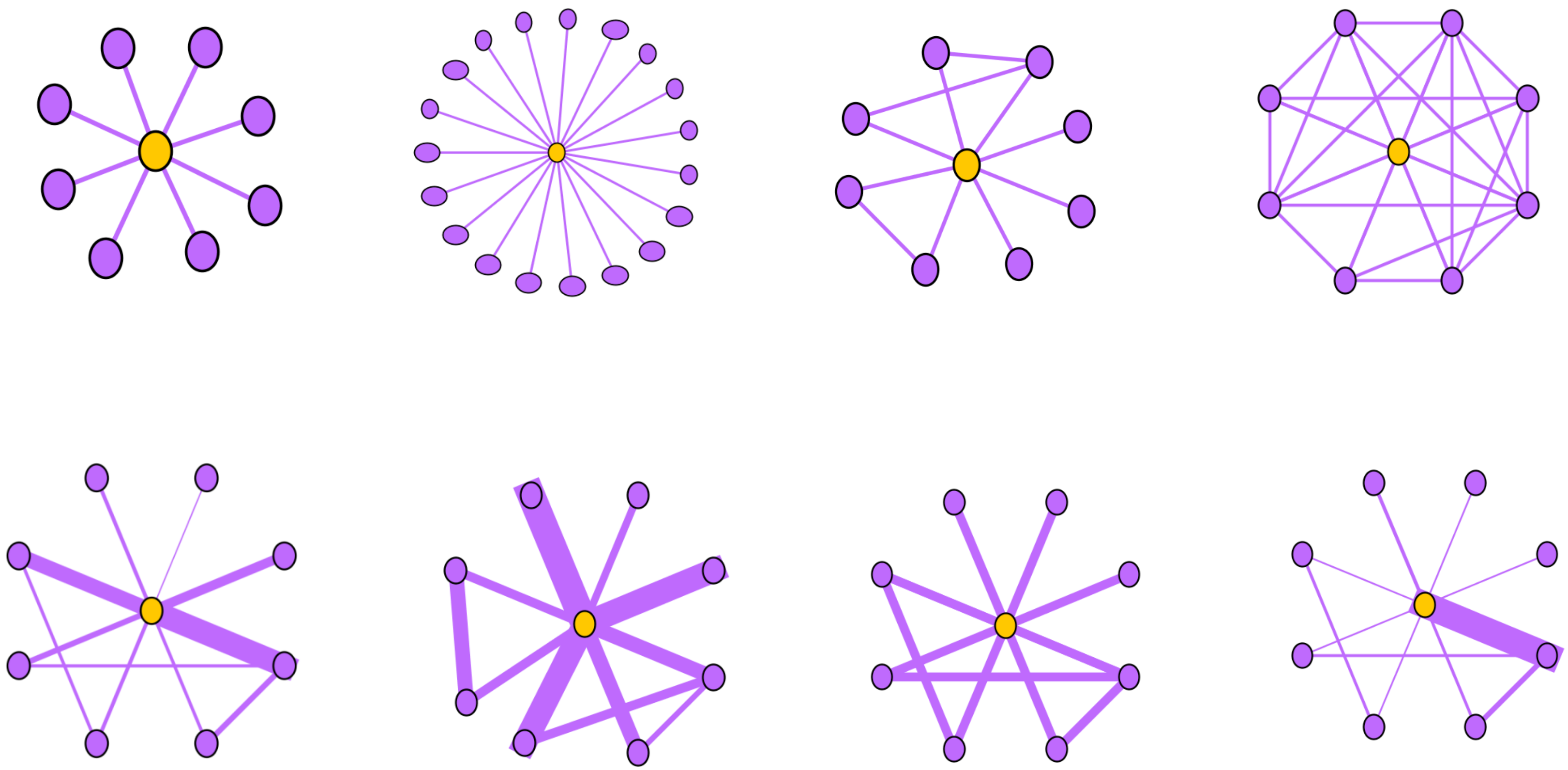
   → regularities

3) Detect anomalies:

   →"distance" to patterns

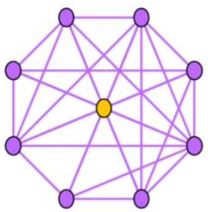
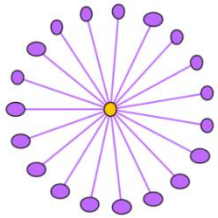


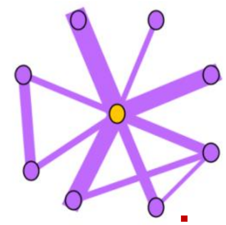
What is odd?



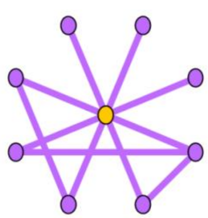
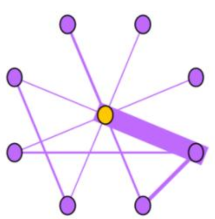
Which features to compute?

- N_i: number of neighbors (degree) of ego *i*
- E_i: number of edges in egonet *i*

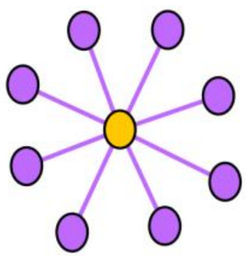


- W_i: total weight of egonet *i*
- $\lambda_{w,i}$: principal eigenvalue of the weighted adjacency matrix of egonet *i*

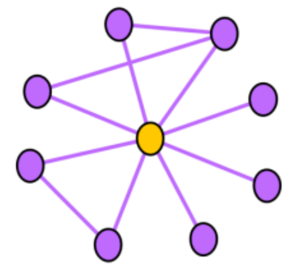



Weighted principal eigenvalue

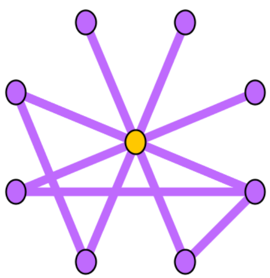
details



$$\lambda_{w,i} = \sqrt{N} = \sqrt{E} = \sqrt{W}$$

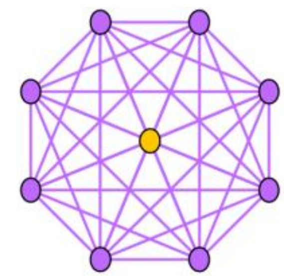


$$\lambda_{w,i} > \sqrt{N}$$
$$\propto \sqrt{E}, \sqrt{W}$$



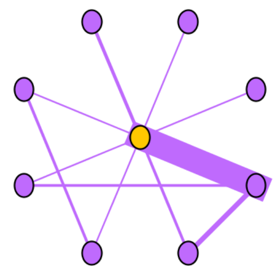
$$\lambda_{w,i} \propto \sqrt{W}$$

$$\lambda_{w,i} = N \approx \sqrt{W}$$



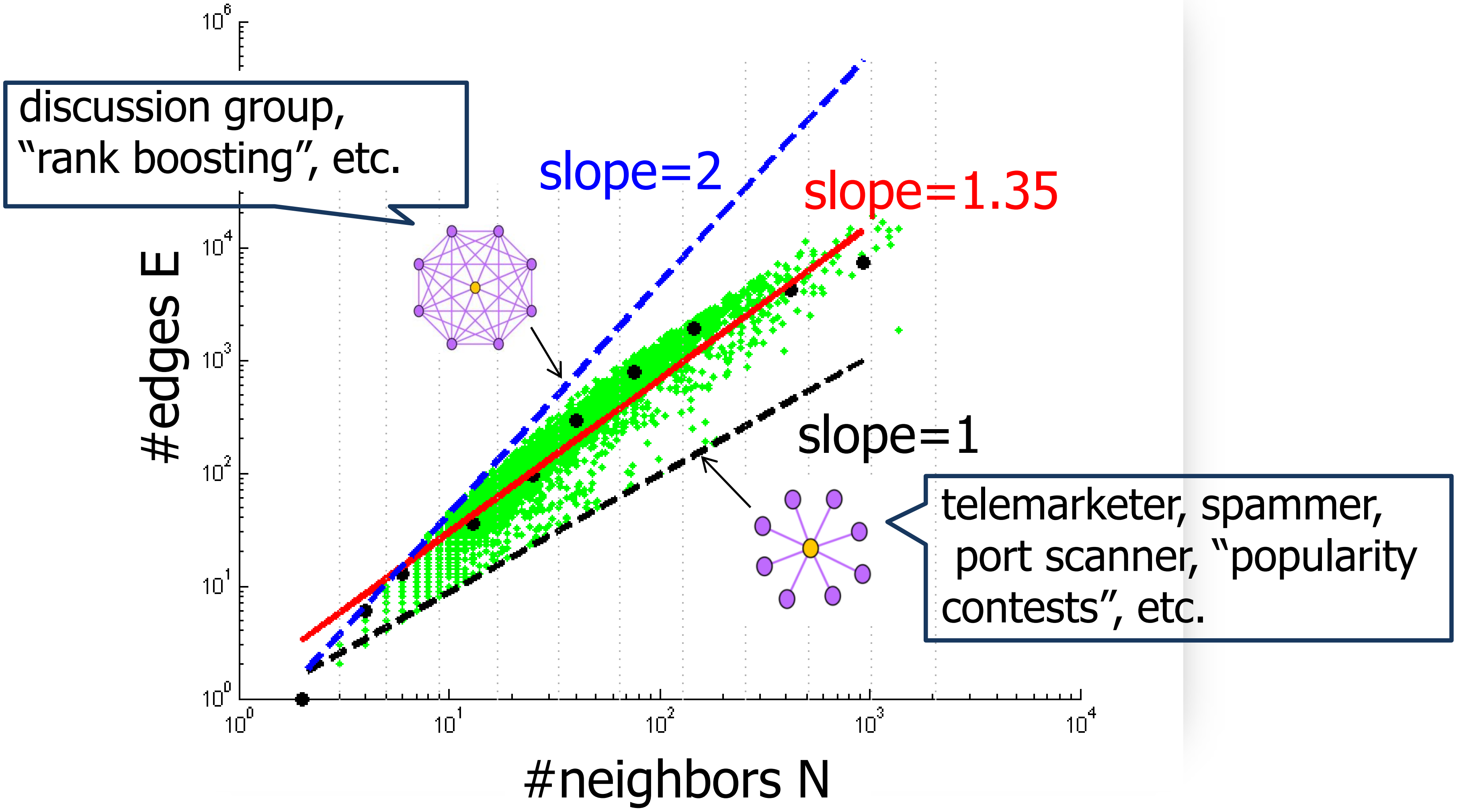
$$\lambda_{w,i} = W$$

$$\lambda_{w,i} \approx W$$

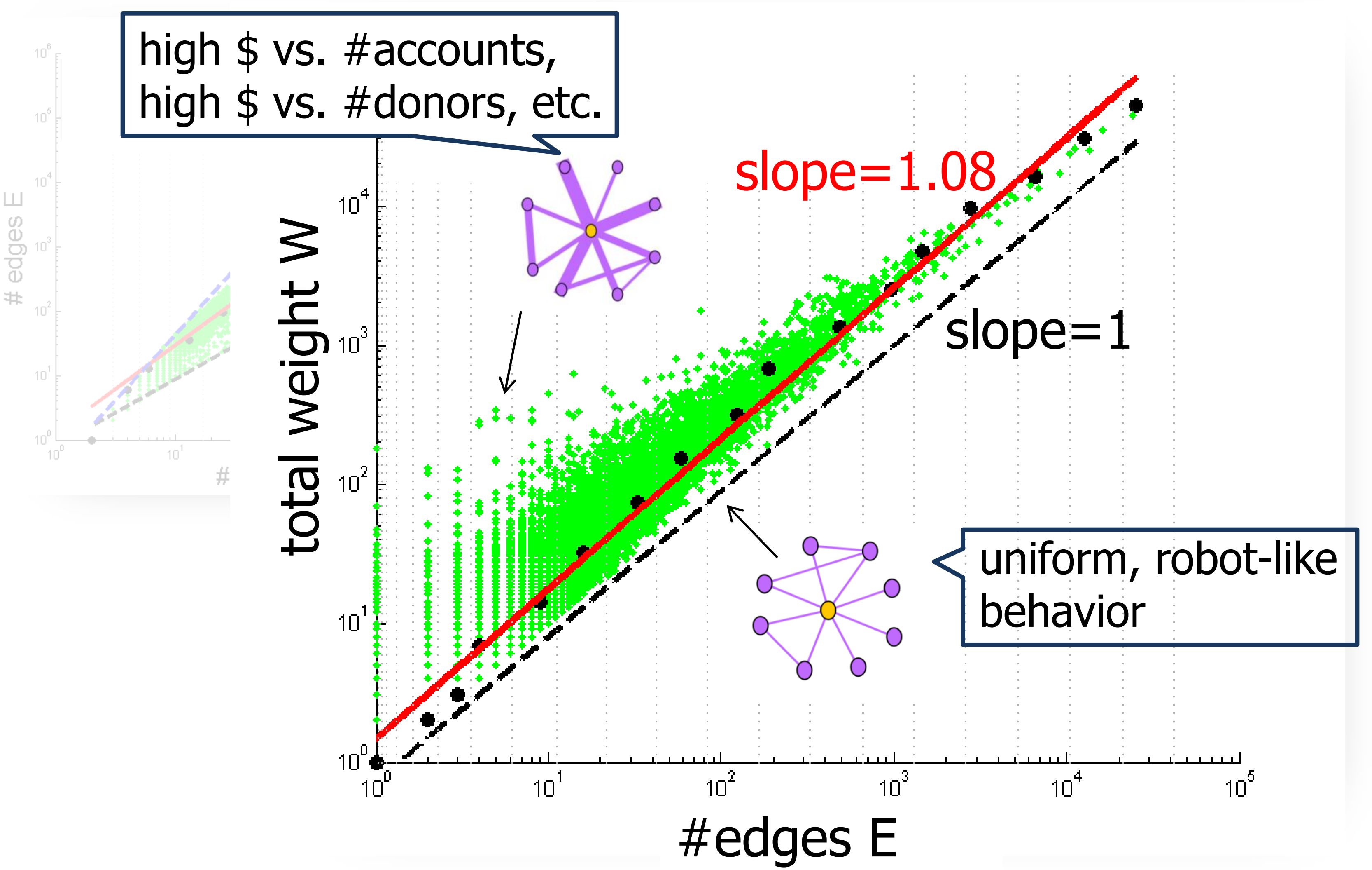


N: #neighbors, W: total weight

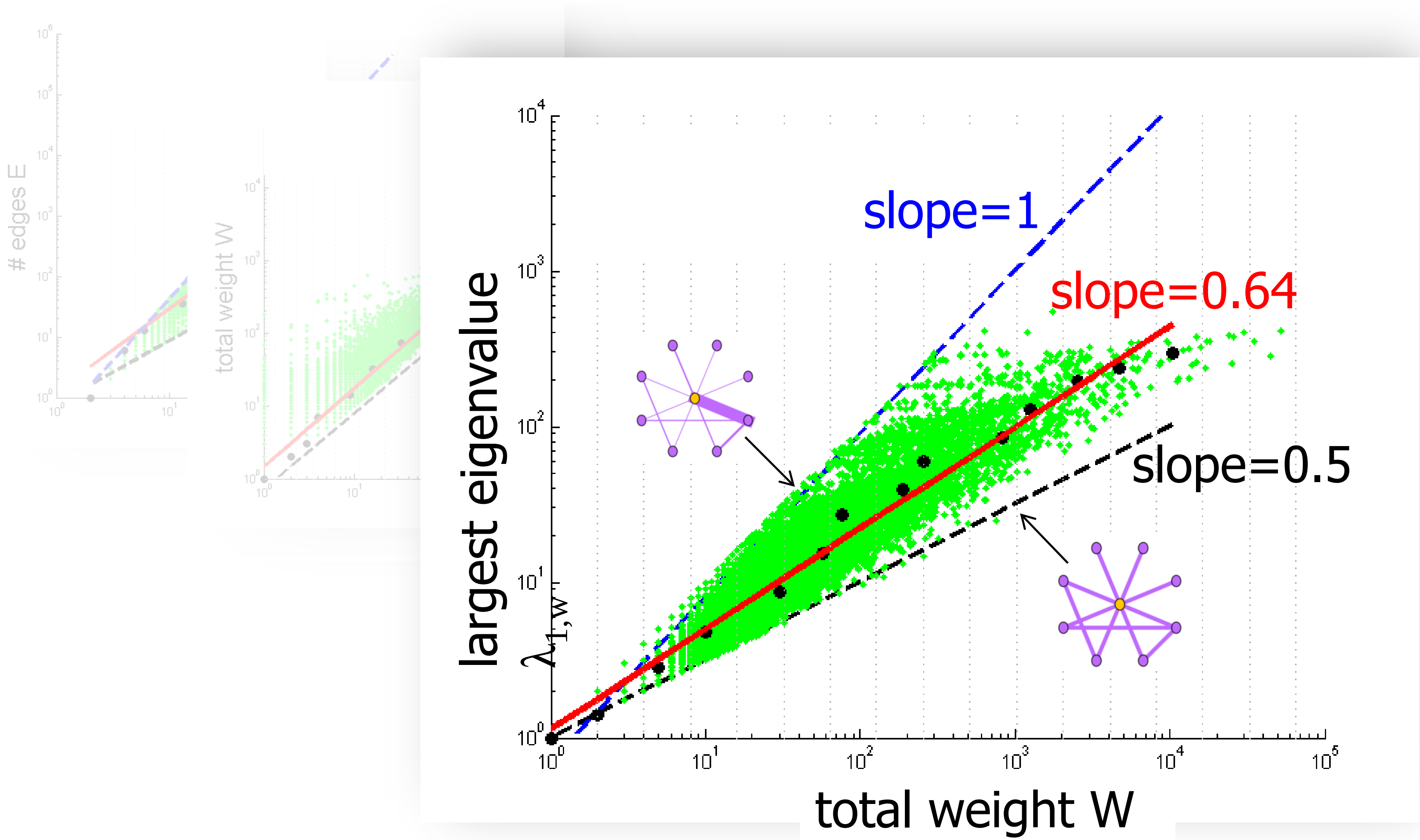
OddBall: pattern#1



OddBall: pattern#2



OddBall: pattern#3

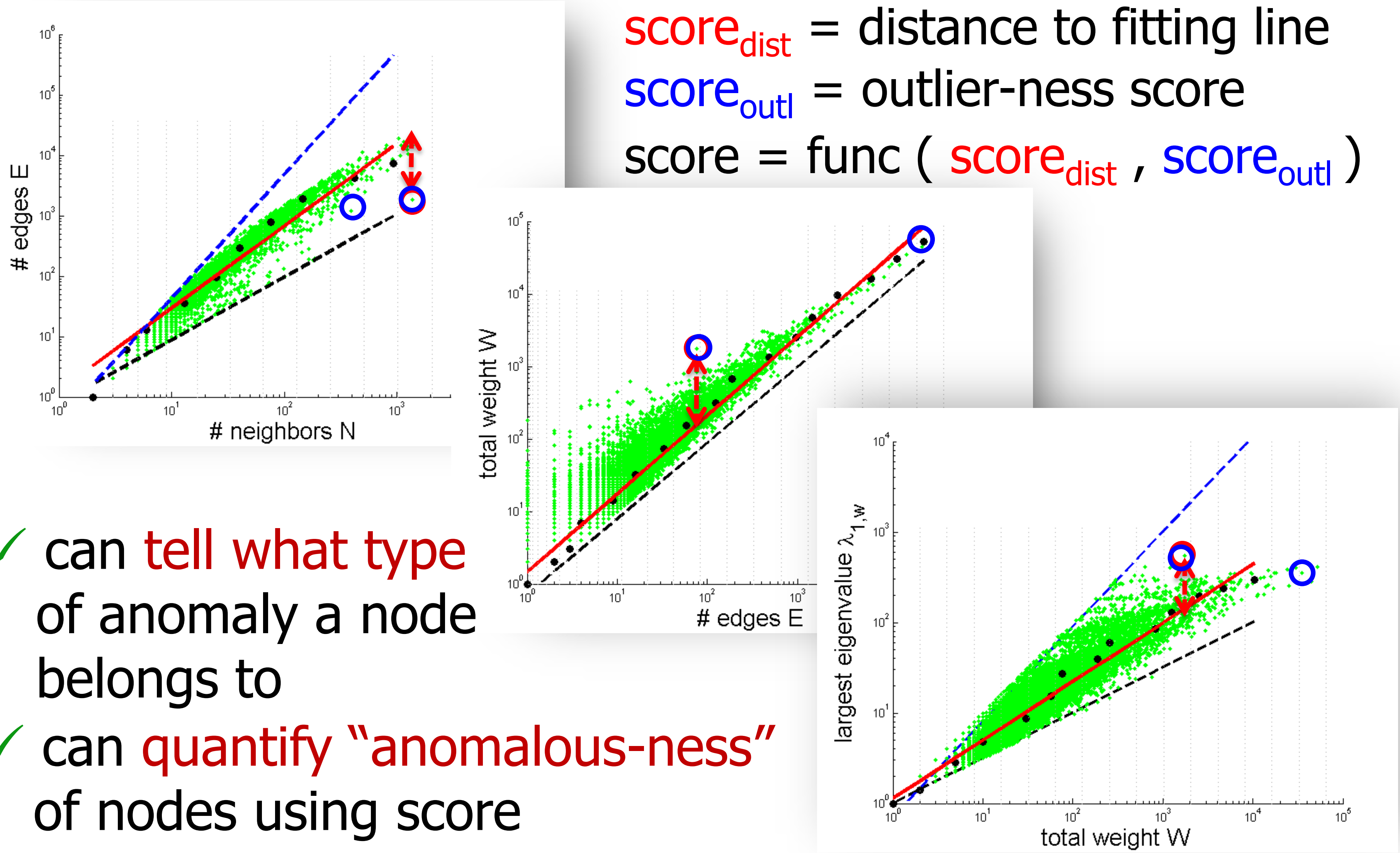


OddBall: anomaly detection

score_{dist} = distance to fitting line

score_{outl} = outlier-ness score

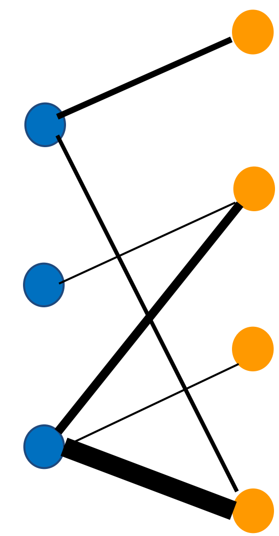
score = func (score_{dist} , score_{outl})



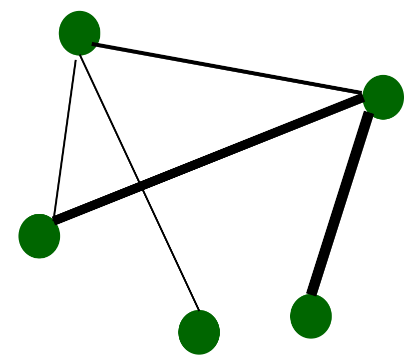
- ✓ can tell what type of anomaly a node belongs to
- ✓ can quantify "anomalous-ness" of nodes using score

OddBall: datasets

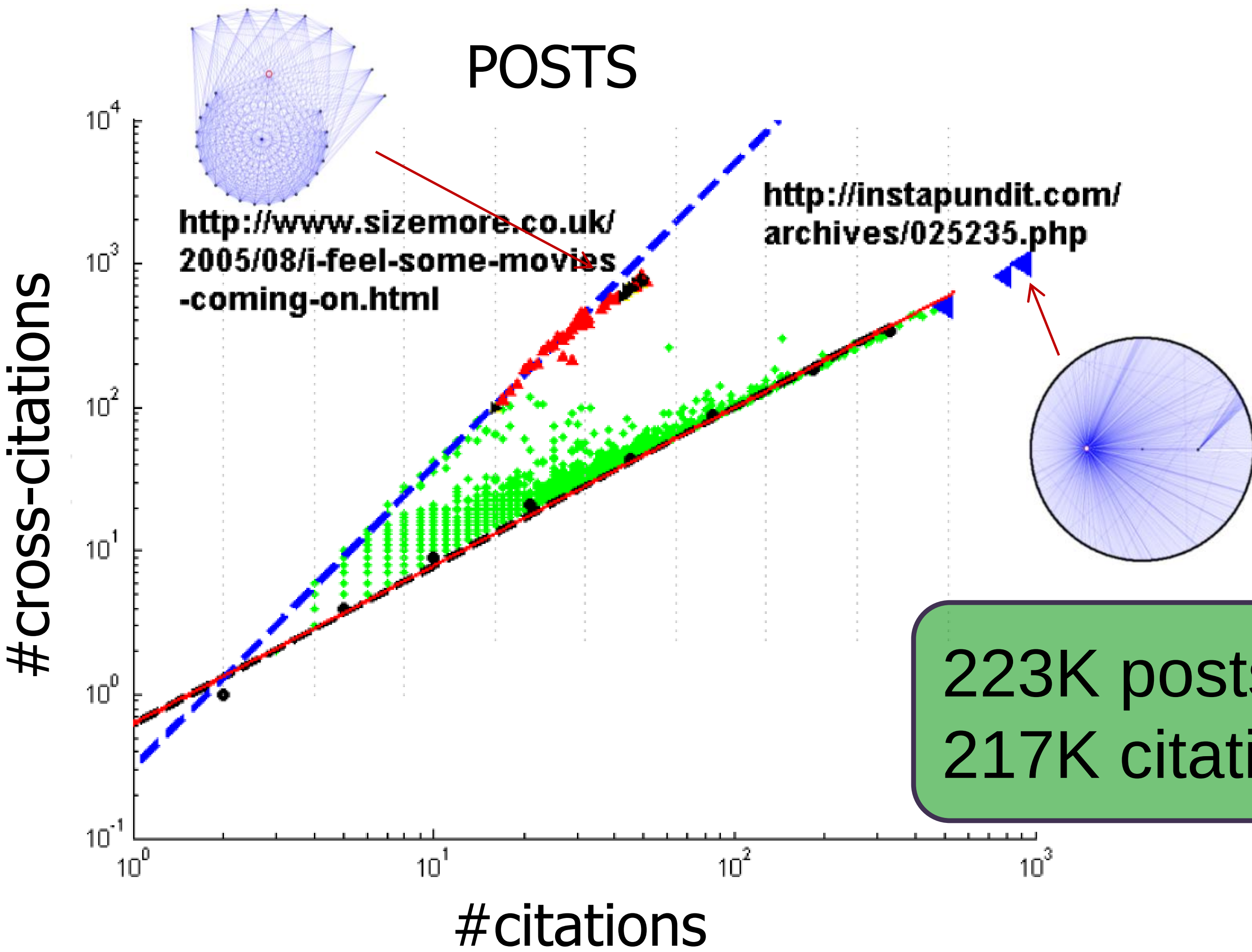
Bipartite graphs:	|V|	|E|
1. **FEC Don2Com**	1.6M	2M
2. **FEC Com2Cand**	6K	125K
3. **DBLP Auth2Conf**	21K	1M



Unipartite graphs:	|V|	|E|
4. **BlogNet**	27K	126K
5. **PostNet**	223K	217K
6. Enron	36K	183K
7. AS peering	11K	8K

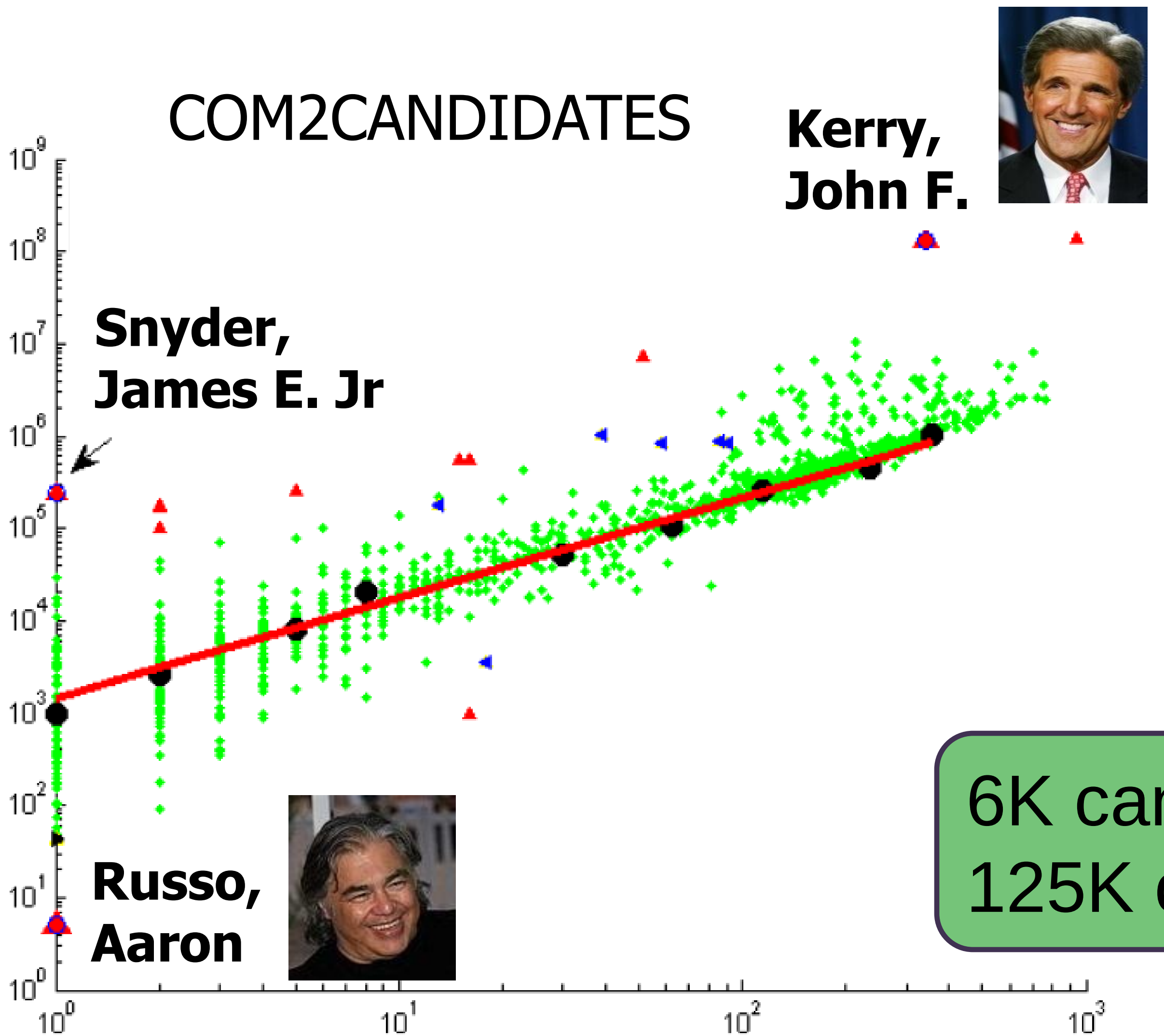


OddBall at work (Posts)



223K posts
217K citations

OddBall at work (FEC)

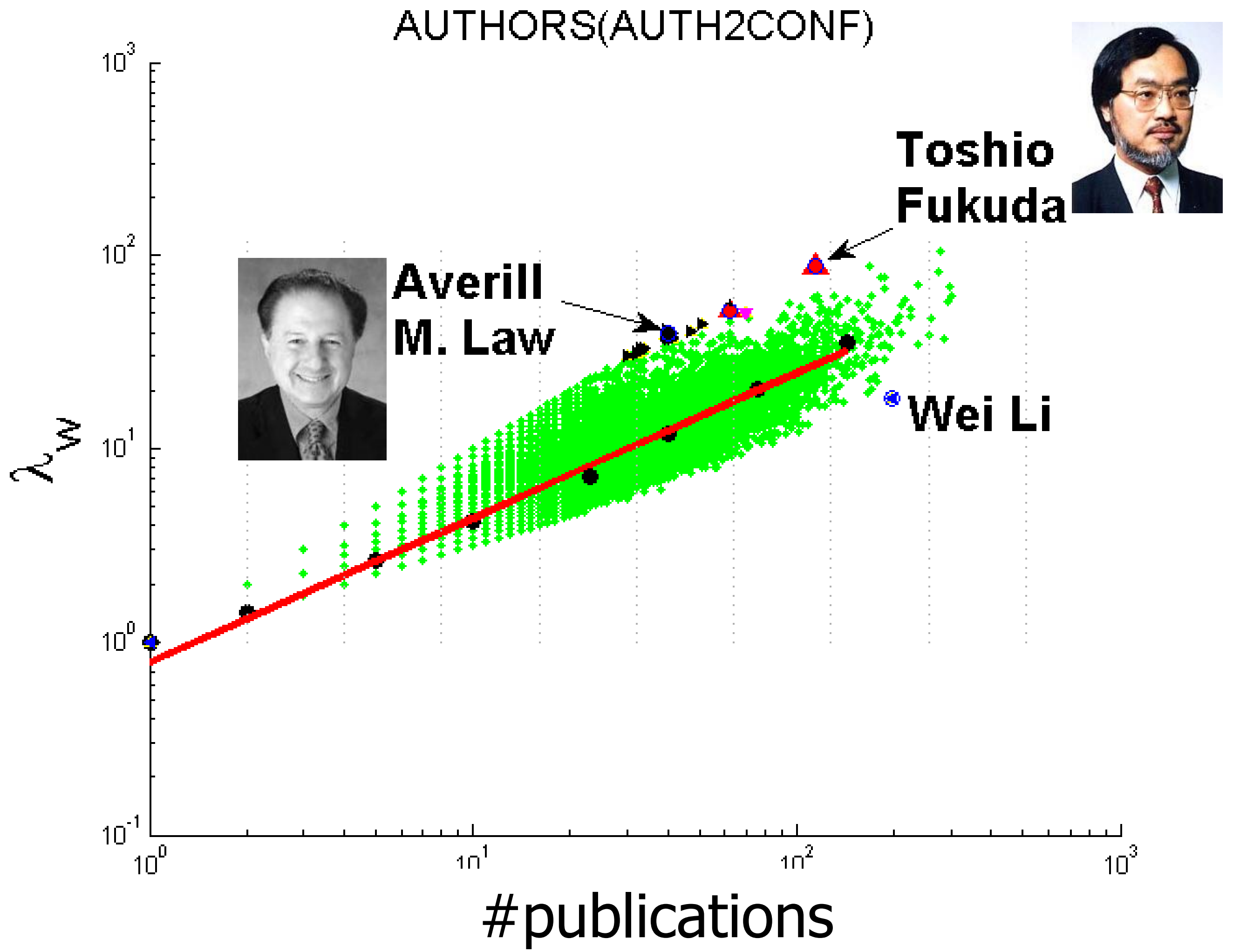


$

#checks

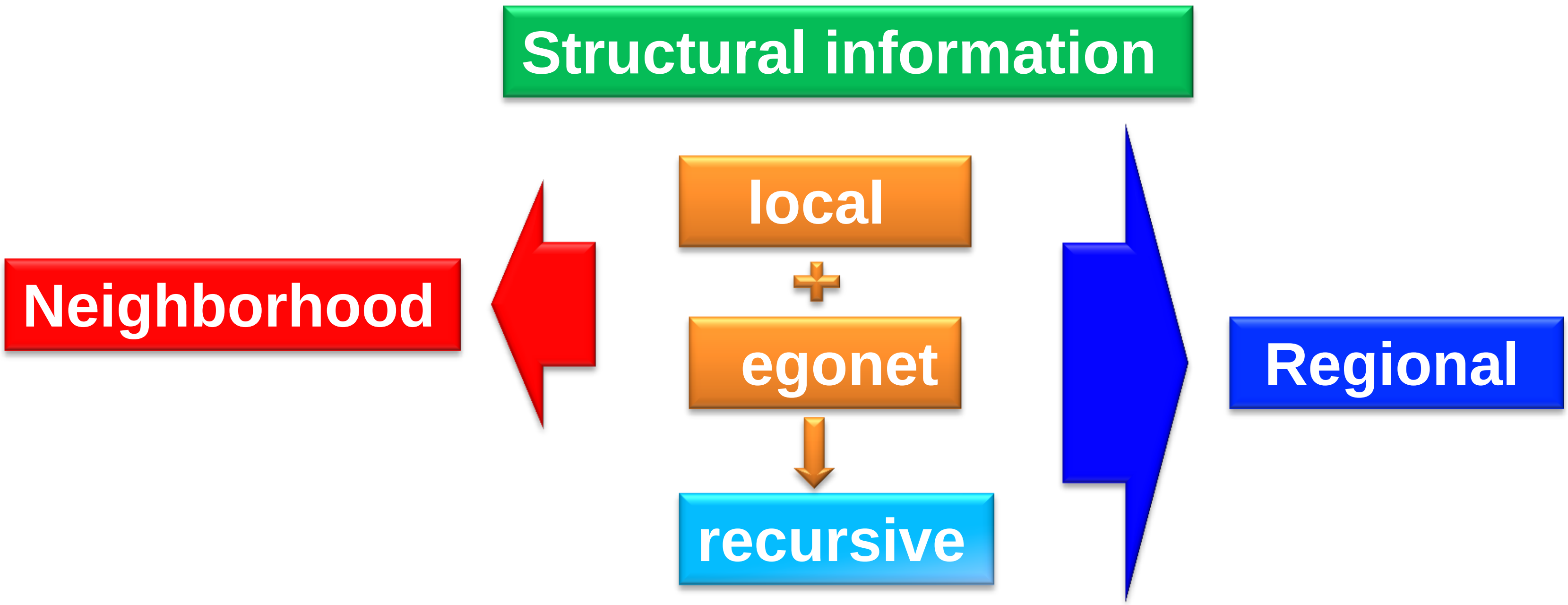
6K candidates
125K checks

OddBall at work (DBLP)



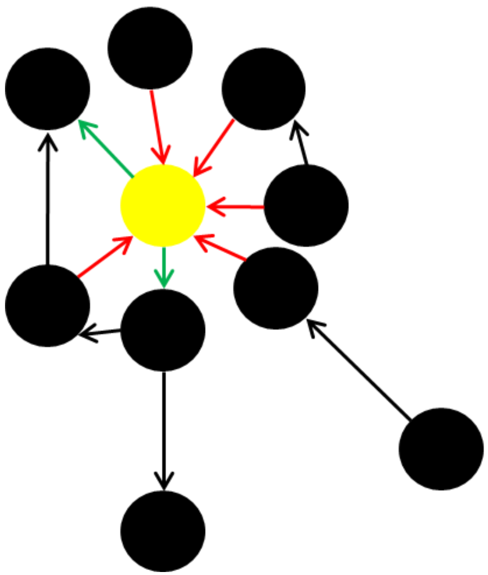
Recursive structural features

- Main idea: recursively combine "local" (node-based) and neighbor (egonet-based) features
  - **Recursive feature:** any aggregate computed over any feature (including recursive) value among a node's neighbors



Recursive structural features

local



in- and out-degree, **weighted** versions

within-, incoming-, outgoing-*egonet* edges, **weighted** versions

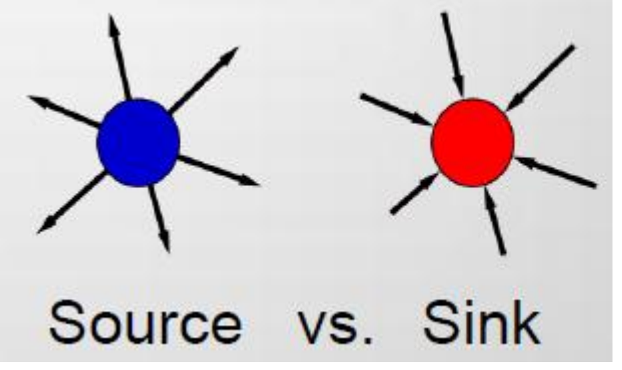
aggregate feature over neighbors

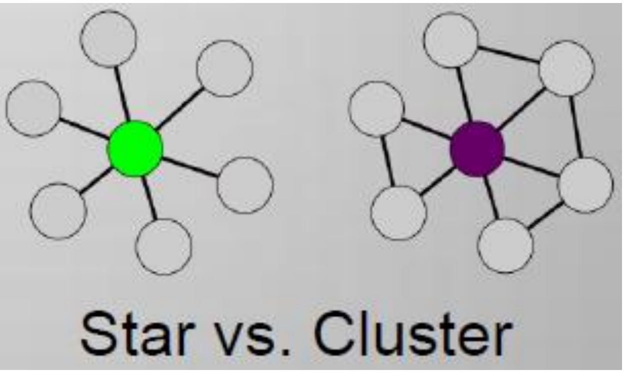
e.g. max/min/avg degree

(1 + 1 + 2 + 0 + 1 + 0 + 1)/7 = 0.86

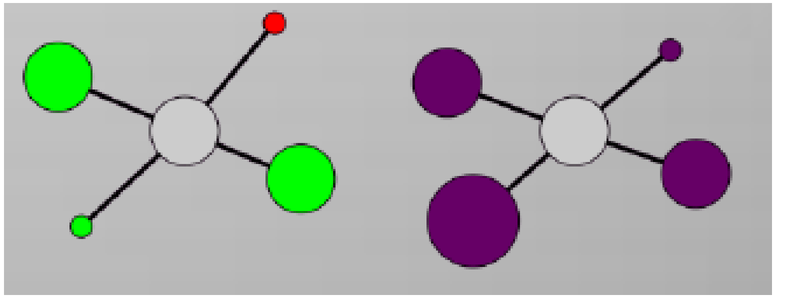
Recursive structural features

- Neigborhood features
  - captures node connectivity



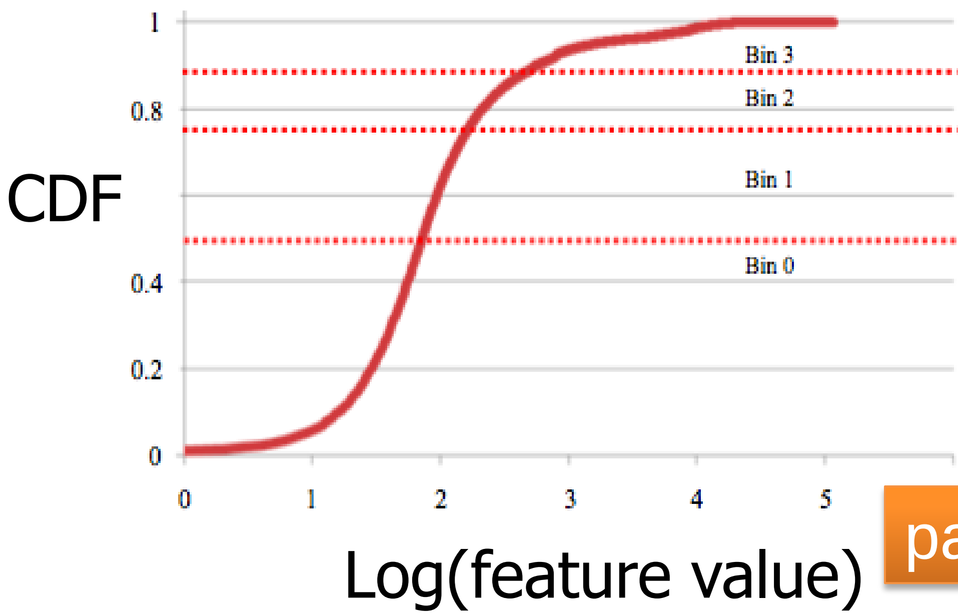


- Regional features
  - captures “kinds” of neighbors



Computing recursive features

details



recursive features

vertical logarithmic binning of size p

bin feature (integer)

not disagree at >s nodes

paired features (s-friend)

replace each CC in s-friend graph by single feature

retain simpler features

i.e. generated in fewer iterations

retained features from each iteration

repeat until no pruning

Prune highly correlated features

Recursive structural features

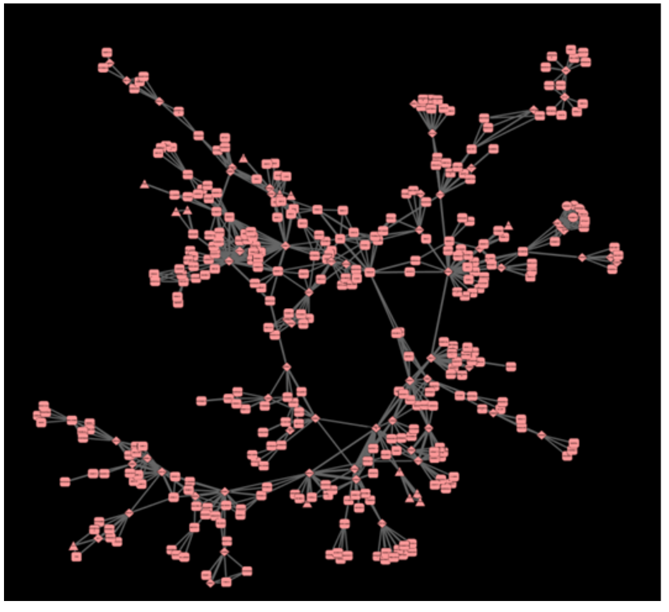
Advantages:

- Capturing regional (behavioral) information in large graphs
- Feature construction linear in graph size

Notes:

- Aggregates only for numerical features
- Parameters p, s for binning and pruning

ReFeX: Recursive Feature eXtraction



- Recursive features proved effective in transfer learning, identity resolution
  (yet to be studied for anomaly detection)

Sun et al. '05

Anomalies in Bipartite Graphs

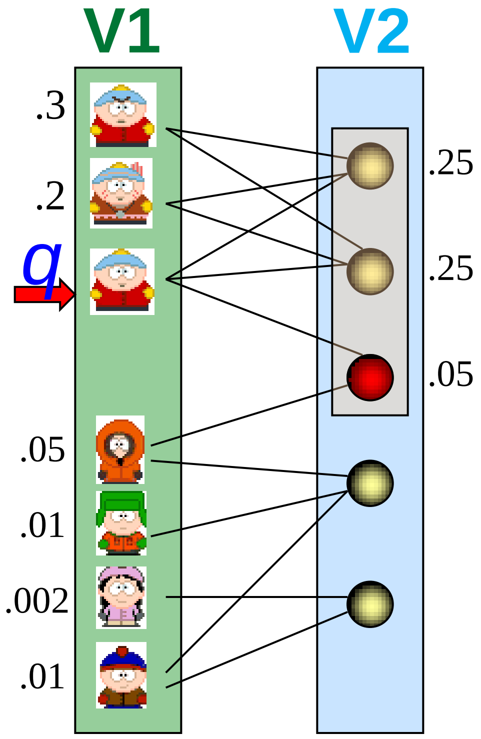
- Problem:

Q1. **Neighborhood formation (NF)**

- Given a query node q in V_1, what are the relevance scores of all the nodes in V_1 to q ?

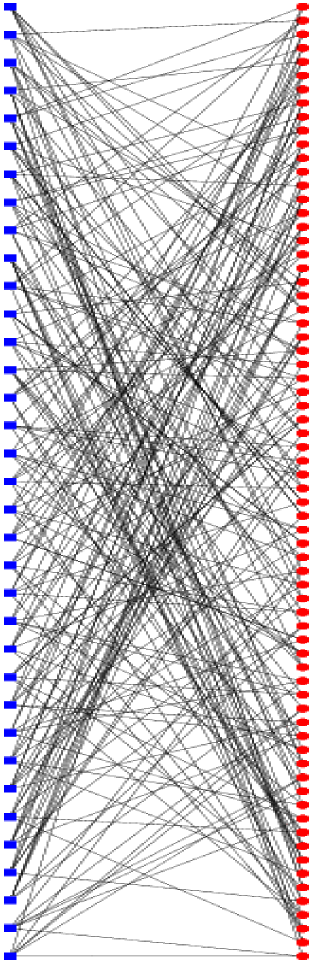
Q2. **Anomaly detection (AD)**

- Given a query node q in V_1, what are the normality scores for nodes in V_2 that link to q ?



Applications of problem setting

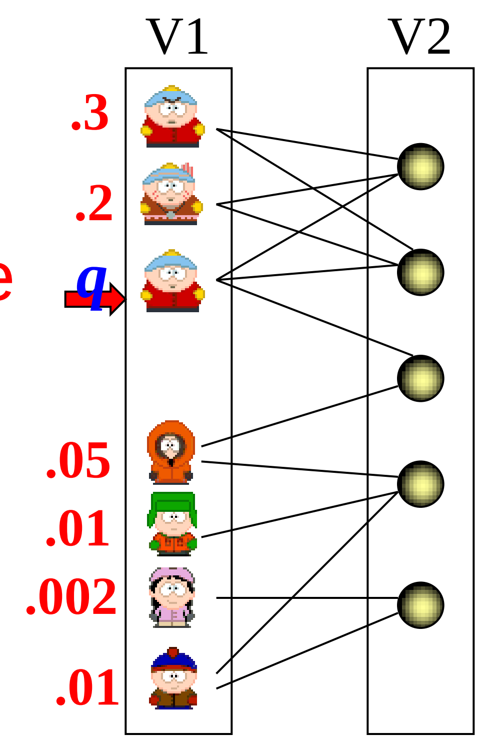
- Publication network
  - (similar) authors vs. (unusual) papers
- P2P network
  - (similar) users vs. ("cross-border") files
- Financial trading network
  - (similar) stocks vs. (cross-sector) traders
- Collaborative filtering
  - (similar) users vs. ("cross-border") products



Sun+ICDM'05
modified with permission

1) Neighborhood formation

- Main idea:
  - Random-Walk-with Restart from q
  - Steady-state V1 prob.s as relevance

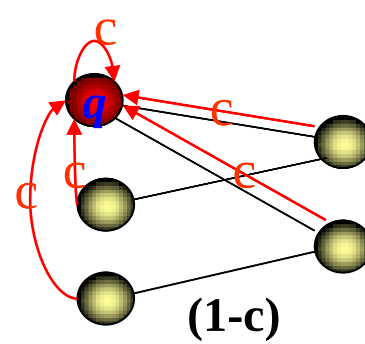


  - (1) Construct transition matrix P

$$P(a,b) = \begin{cases} \frac{1-c}{\text{outdeg}(a)} & \text{if } (a,b) \in E \\ 0 & \text{if } (a,b) \notin E \end{cases}$$

  - (2) Fly-back prob. c to q
  - (3) Solve for steady state

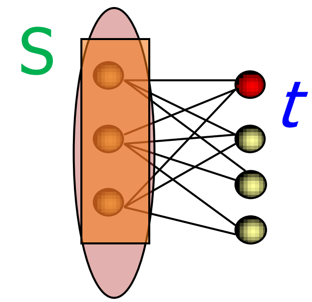
$$\vec{u}_a^{(t+1)} = P\,\vec{u}_a^{(t)} + c\vec{q}$$



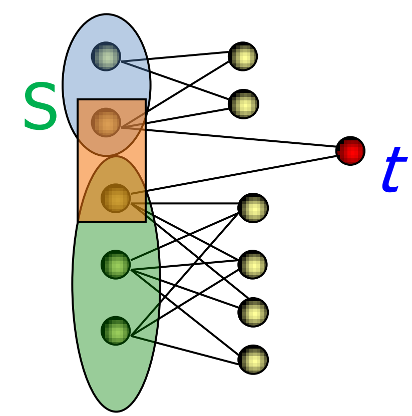
Approx: RWR on graph **partition** containing q

2) Anomaly detection

- Main idea:
  - Pairwise “normality” scores of neighbors(t)
  - Function of (e.g. avg) pair-wise scores

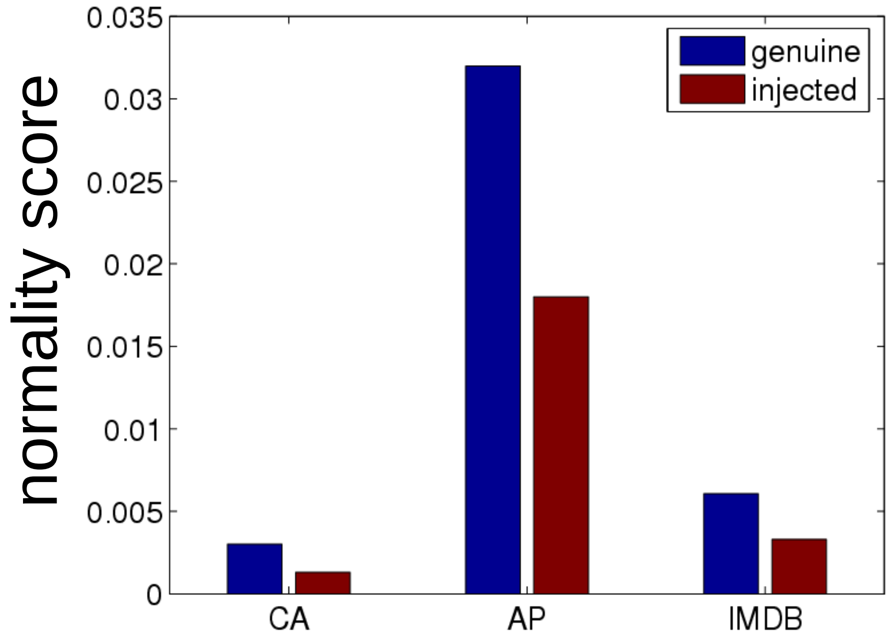


  - (1) Find set S of nodes connected to t
  - (2) Compute |S|x|S| normality matrix R
    - asymmetric, diagonal reset to 0
  - (3) Apply score function f(R)
    - e.g. f(R) = mean(R)



Experiment

- 3 real datasets
  - DBLP Conf-Auth
  - DBLP Auth-Paper
  - IMDB movie-actor



- Randomly inject 100 nodes, each with k (avg. degree) edges (biased towards high-degree nodes)
- No qualitative results on real nodes ranked top

Tong et al. '11

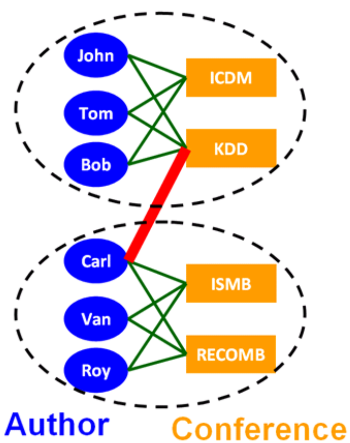
Graph Anomalies by NNrMF

- Low-rank adjacency matrix factorization of a (sparse) graph reveals communities and anomalies

Low-rank matrices   Residual matrix

Graph → Adj. Matrix A → $A = F \times G + R$

community   anomalies



Non-negativity constraints

- For improved interpretability

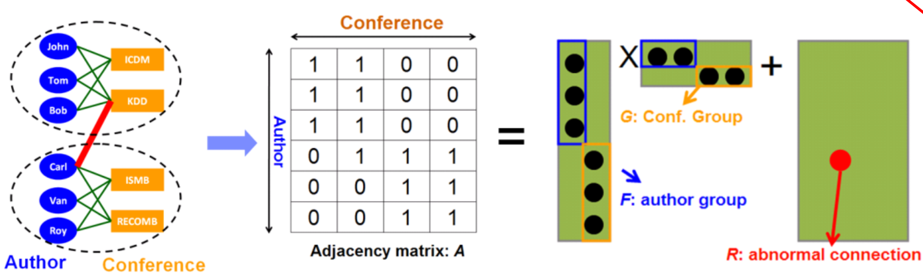
- A Typical Procedure:

community

Graph → Adjacency Matrix *A* → $A = F \times G + R$

anomalies

- An Example



1	1	0	0
1	1	0	0
1	1	0	0
0	1	1	1
0	0	1	1
0	0	1	1

Adjacency matrix: *A*

Interpretation by Non-negativity

Non-negative Matrix Factorization
$F >= 0$; $G >= 0$
(for community detection)

Non-negative Residual Matrix Factorization
$R(i,j) >= 0$; for $A(i,j) > 0$
(for anomaly detection)

Optimization formulation

Common in Matrix Factorization

$$\operatorname{argmin}_{\mathbf{F},\mathbf{G}} \sum_{i,j,\ \mathbf{A}(i,j)>0} (\mathbf{A}(i,j) - \mathbf{F}(i,:)\mathbf{G}(:,j))^2$$

$$\text{s.t.} \quad \text{for all } \mathbf{A}(i,j) > 0:$$

Non-negative residual →

$$\mathbf{F}(i,:)\mathbf{G}(:,j) \leq \mathbf{A}(i,j)$$

- Q: How to find 'optimal' **F** and **G**?
  - D1: Quality ←→ C1: objective non-convex
  - D2: Scalability ←→ C2: large graph size

Optimization: batch

details

- Basic Idea 1: Alternating

$$\text{argmin}_{\mathbf{F},\mathbf{G}} \sum (\mathbf{A}(i,j) - \mathbf{F}(i,:)\mathbf{G}(:,j))^2$$

Not convex w.r.t. *F* and *G*, *jointly*
But convex if fixing either *F* or *G*

- Basic Idea 2: Separation

argmin$_G$ $\displaystyle\sum_{i,j,\ \mathbf{A}(i,j)>0} (\mathbf{A}(i,j) - \mathbf{F}(i,:)\mathbf{G}(:,j))^2$

s.t..

for all $\mathbf{A}(i,j) > 0$:

$\mathbf{F}(i,:)\mathbf{G}(:,j) \leq \mathbf{A}(i,j)$

For each *j*

argmin$_G$ $\displaystyle\sum_{i,\ \mathbf{A}(i,j)>0} (\mathbf{A}(i,j) - \mathbf{F}(i,:)\mathbf{G}(:,j))^2$

s.t.. for all $\mathbf{A}(i,j) > 0$:

$\mathbf{F}(i,:)\mathbf{G}(:,j) \leq \mathbf{A}(i,j)$

Standard Quadratic Programming

Overall Complexity: Polynomial

Optimization: incremental

details

- Basic Idea 0: Recursive
- Basic Idea 1: Alternating

$$\operatorname{argmin}_{\mathbf{f},\mathbf{g}} \sum_{i,j,\ \mathbf{A}(i,j)>0} (\mathbf{A}(i,j) - \mathbf{f}(i)\mathbf{g}(j))^2$$

For each *j*

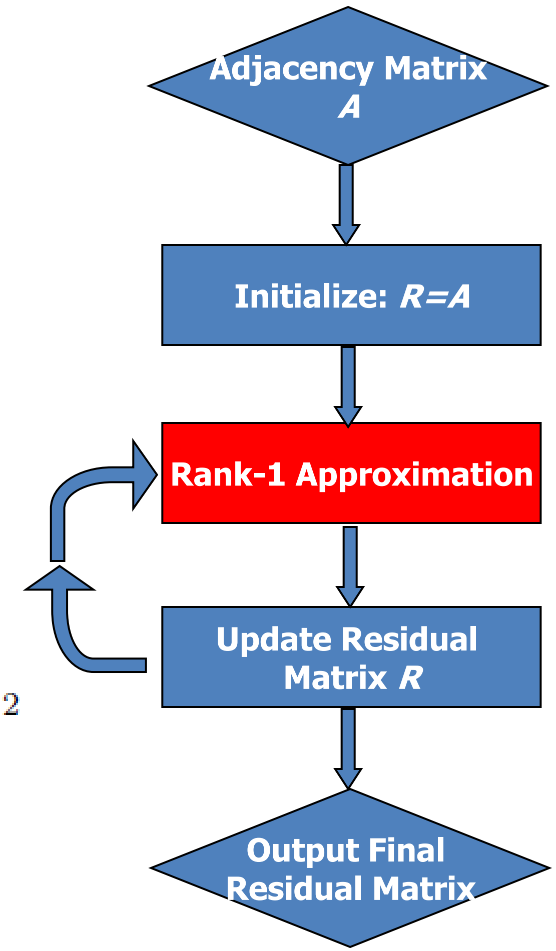
- Basic Idea 2: Separation

QP for a single variable w/ boundary constrains

Solved in constant time

$$\operatorname{argmin}_{\mathbf{g}_j} \sum_{i,\ \mathbf{A}(i,j)>0} (\mathbf{A}(i,j) - \mathbf{f}(i)\mathbf{g}(j))^2$$

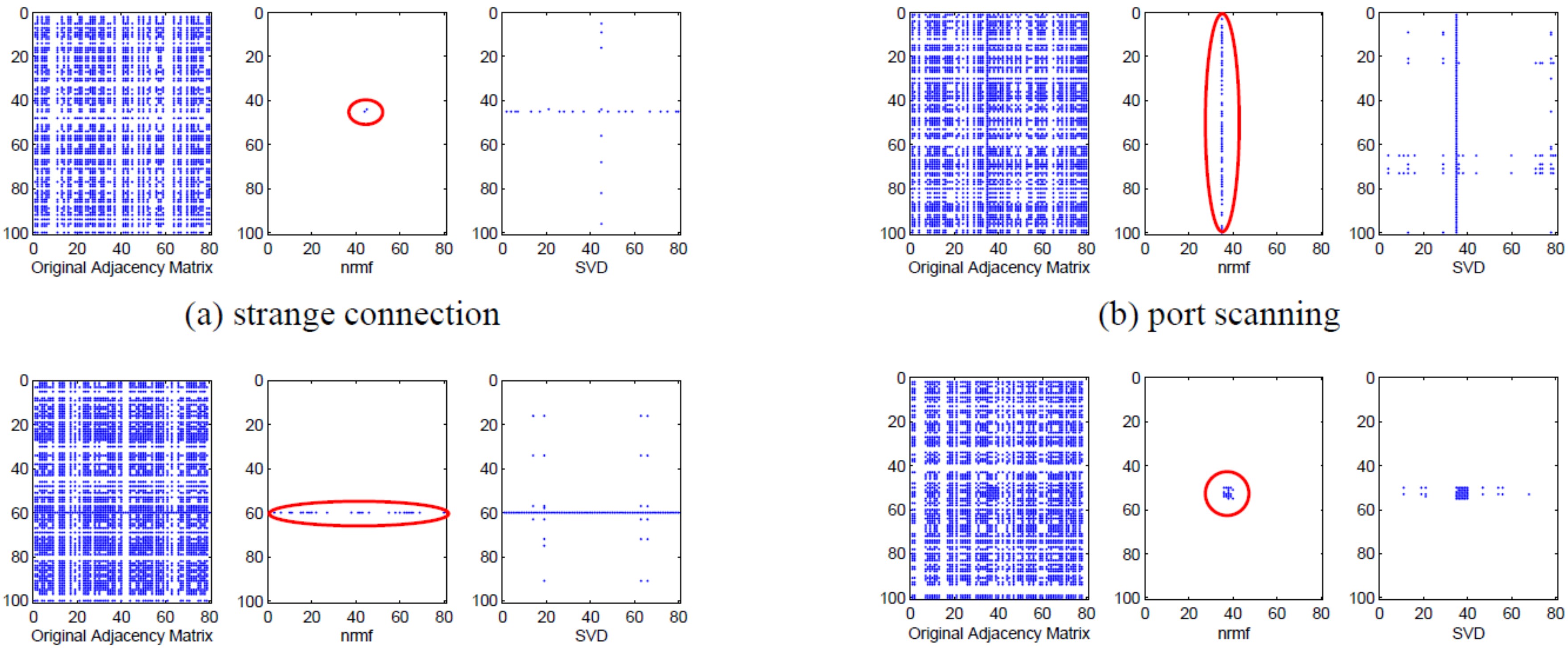
$$\text{s.t.} \quad \text{for all } \mathbf{A}(i,j) > 0: \quad \mathbf{f}(i)\mathbf{g}(j) \leq \mathbf{A}(i,j)$$



Overall Complexity: Linear wrt # of edges

Experiments

- NNrMF can spot 4 types of anomalies



(a) strange connection

(b) port scanning

(c) ddos

(d) bipartite core

NNrMF residuals

SVD residuals

(top-k edges)

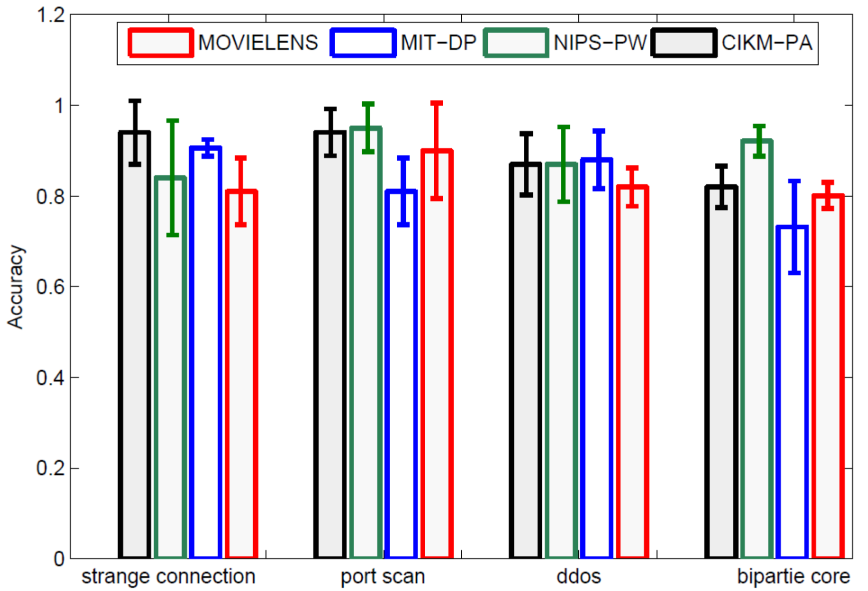
Tong+SDM'11 modified with permission

Experiments

- 4 real datasets, with injected anomalies

Effectiveness

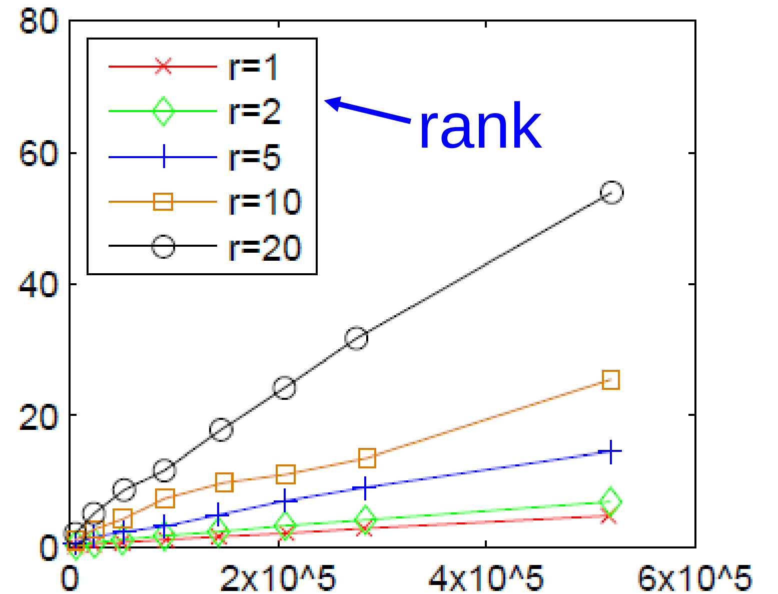
Accuracy



Anomaly Type

Efficiency

Wall-clock time (s)



# of edges

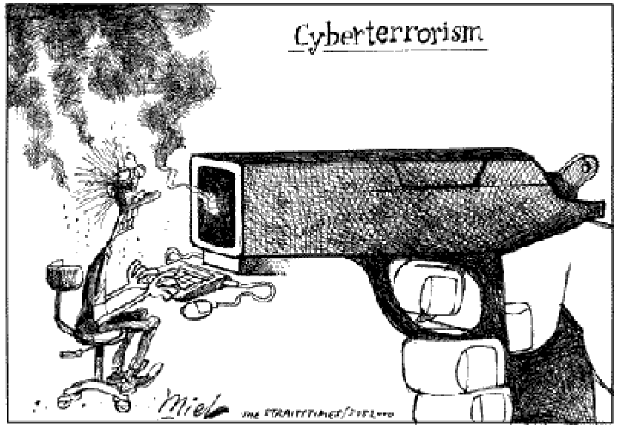
Tong+SDM'11
modified with permission

Ding et al. '12

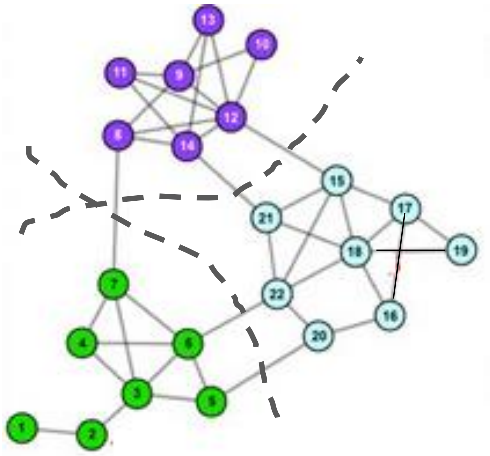
Intrusion as (Anti)social Communication

- Problem:

Q. How to detect malicious attacks in computer networks?



- Main insight for intrusion:
  - entering a community to which one doesn't belong
  - look for communication that does not respect **community boundaries**

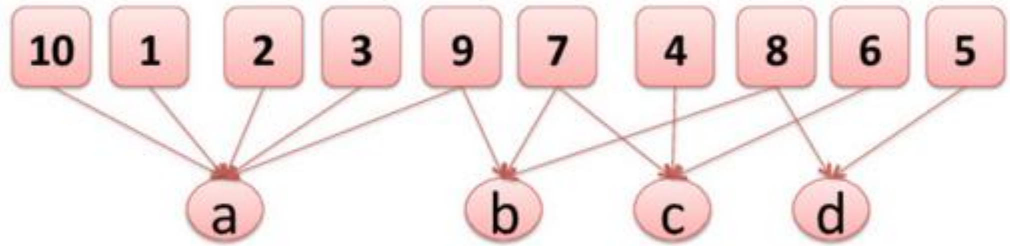


Problem formulation

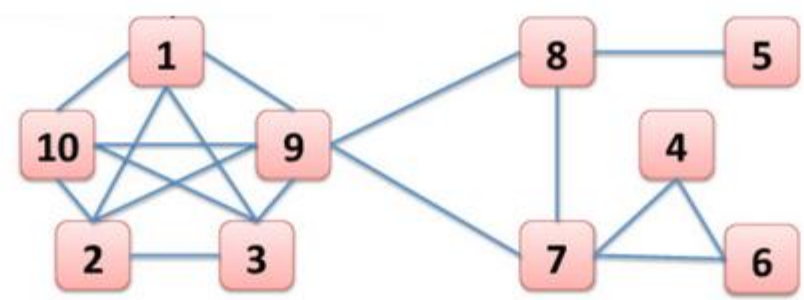
- Network representation as a bipartite graph

Source IPs ⟶

Dest. IPs ⟶



  - Source and destination IPs may overlap
- One mode projection G_P: connect two source IPs with at least 1 common neighbor
- Alternative G_W: weigh by correlation coefficient



Intrusion data with ground truth

- Data: netflow traffic
  - from a large European ISP
  - 2 weeks data in 2007: source IP, dest IP, start/end time, number of bytes/packets sent
  - Ground truth: traffic sources that attempted an intrusion as recorded by Dshield*
    - known IPs sending malicious or unwanted traffic

* http://www.dshield.org/

Detection methods

- **Community detection:** Standard community detection methods fail to distinguish known IPs from communities

Clauset, Newman, Moore '04

Size of Cluster	# of Clusters	# of DShields
6784	1	158
986	1	1
8 to 243	10	0
≤ 7	56	2
Total	68	161

- **Cut-vertices:**

Iteratively remove cut-vertices

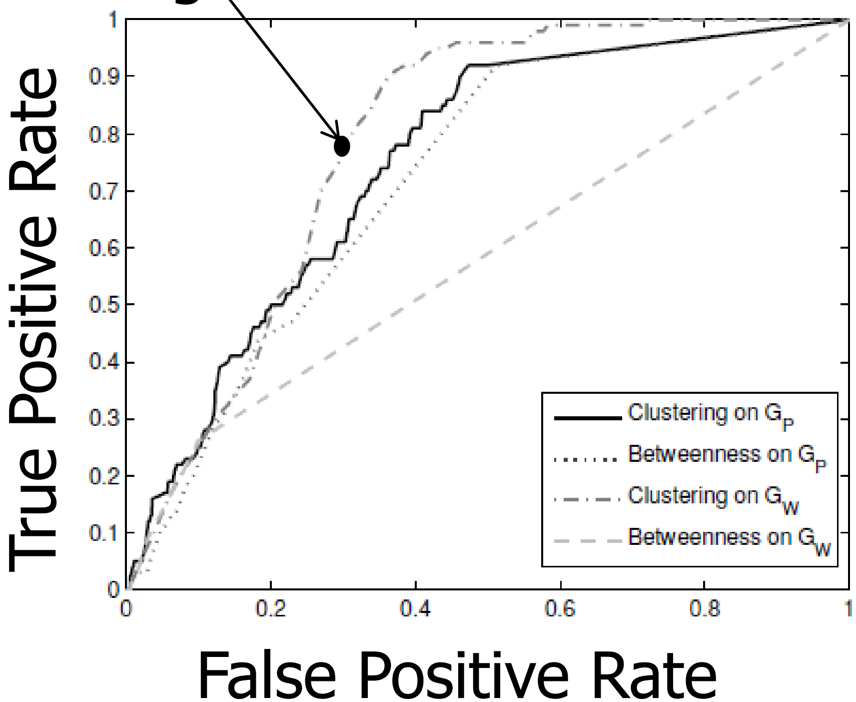
  - 6.6% of cut-vertices are Dshields (randomization –-randomly reassign Dshield nodes—yields significance; (1-2.2%) at 0.05)

→ **Clustering** and **betweenness** are discriminative

Experiments

- Malicious if clustering/betweenness below/above threshold

for a given threshold



	Mean(AUC)	SE(AUC)
Clustering on G_P	0.7440	0.0103
Betweenness on G_P	0.7180	0.0084
Clustering on G_W	0.7625	0.0080
Betweenness on G_W	0.5621	0.0034

- Clustering gives better discrimination
- GW does not provide much improvement over GP

Part I: References (plain graphs)

Feature mining

Community mining

- L. Akoglu, M. McGlohon, C. Faloutsos. OddBall: Spotting Anomalies in Weighted Graphs. PAKDD, 2010.
- K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, C. Faloutsos. It's Who You Know: Graph Mining Using Recursive Structural Features. KDD, 2011.
- J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. ICDM, 2005.
- Hanghang Tong, Ching-Yung Lin: Non-Negative Residual Matrix Factorization with Application to Graph Anomaly Detection. SDM, pages 143-153, 2011.
- Q. Ding, N. Katenka, P. Barford, E. Kolaczyk, and M. Crovella. Intrusion as (Anti)social Communication: Characterization and Detection. KDD, 2012.

Part I: Outline

- Overview: Outliers in **clouds of points**
  - Outliers in **numerical** data points
    - distance-based, density-based, …
  - Outliers in **categorical** data points
    - model-based

- Anomaly detection in **graph data**
  - Anomalies in unlabeled, **plain** graphs
  - Anomalies in node-/edge-labeled, **attributed** graphs

Coffee break…