1. Subgraph Analysis
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

2. Propagation Methods

3. Latent Factor Models
What is a subgraph?
What is a subgraph?

Subset of nodes and the edges between them
What are some useful subgraphs?

Largest dense subgraph (Greatest average degree)
What are some useful subgraphs?

Ego-network: the subgraph among a node and its neighbors
What are some useful subgraphs?

Graph queries:
find subgraphs of particular pattern
What are some useful subgraphs?

Graph queries: find subgraphs of particular pattern
What are some useful subgraphs?

Graph queries:
find subgraphs of particular pattern
What are some useful subgraphs?

Graph queries:
find subgraphs of particular pattern
What are some useful subgraphs?

Graph queries:
find subgraphs of particular pattern
Subgraphs as submatrices
Subgraphs as submatrices

Rearrange to find dense regions!
Subgraphs as submatrices

Near-Bipartite core
Subgraphs as submatrices
Subgraphs as submatrices

Co-clustering and cross associations:
Partition matrix through clustering rows and columns.

Goal: Each block should have mostly similar cells
1. Subgraph Analysis
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

2. Propagation Methods

3. Latent Factor Models
Ego-net Patterns

Oddball: Spotting anomalies in weighted graphs
Leman Akoglu, Mary McGlohon, Christos Faloutsos
PAKDD 2010
Ego-net Patterns

- $N_i$: 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 $i$
Pattern: Ego-net Power Law Density

\[
E_i \propto N_i^\alpha \\
1 \leq \alpha \leq 2
\]
Oddball: Spotting anomalies in weighted graphs
Leman Akoglu, Mary McGlohon, Christos Faloutsos
PAKDD 2010

Pattern: Ego-net Power Law Weight

\[ W_i \propto E_i^\alpha \]

1 \leq \alpha

slope = 1.08
Pattern: Ego-net Power Law Eigenvalue

\[ \lambda_i \propto W_i^\alpha \]

\[ 0.5 \leq \alpha \leq 1 \]
Using graph patterns to find roles

Useful node features:
• Degree
• Nodes in ego-net
• Edges in ego-net
• Edges leaving ego-net
• Mean of neighbor degree
• Sum of neighbor degree
• Expand recursively…

It's who you know: Graph mining using recursive structural features
K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, C. Faloutsos
KDD 2011
Using graph patterns to find roles

Learn classifier to predict node labels

It’s who you know: Graph mining using recursive structural features
Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, Christos Faloutsos
KDD 2011
Using graph patterns to find roles

<table>
<thead>
<tr>
<th></th>
<th>“Web”</th>
<th>“DNS”</th>
<th>“Peer-to-Peer”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td><img src="web_training.png" alt="Web" /></td>
<td><img src="dns_training.png" alt="DNS" /></td>
<td><img src="peer-to-peer_training.png" alt="Peer-to-Peer" /></td>
</tr>
<tr>
<td>Test</td>
<td><img src="web_test.png" alt="Web" /></td>
<td><img src="dns_test.png" alt="DNS" /></td>
<td><img src="peer-to-peer_test.png" alt="Peer-to-Peer" /></td>
</tr>
</tbody>
</table>

It’s who you know: Graph mining using recursive structural features
Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, Christos Faloutsos

*KDD 2011*
Using graph patterns to find roles

![Graph Pattern Diagram]

It’s who you know: Graph mining using recursive structural features
Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, Christos Faloutsos
KDD 2011
Using graph patterns to find roles

Use graph features to find similar types of behavior:

- Christos Faloutsos & Andrei Broder: tightly knit communities
- Albert-Laszlo Barabasi & Mark Newman: bridge communities
- John Hopcroft and Jon Kleinberg: mainstream
- Lada Adamic and Bernardo Huberman: elongated clusters

RolX: Structural Role Extraction & Mining in Large Graphs
KDD 2012
Using ego-nets to predict engagement

Number of connected components in egonets predicts engagement on Facebook

Structural diversity in social contagion
Johan Ugander, Lars Backstrom, Cameron Marlow, Jon Kleinberg
PNAS 2012
Attributed subgraph patterns

- **SUBDUE**: An algorithm for detecting repetitive patterns (substructures) within (single-attributed) graphs.
- The best substructure is the one that **minimizes**
  \[ F_1(S, G) = DL(G | S) + DL(S) \]
- **G**: Entire graph, **S**: The substructure,
- DL(G|S) is the DL of G after compressing it using S,
- DL(S) is the description length of the substructure.
Friend groups within ego-nets

friends under the same advisor
CS department friends
college friends
‘alters’ $v_i$

‘ego’ $u$

family members
highschool friends

Learning to Discover Social Circles in Ego Networks
Julian McAuley, Jure Leskovec
NIPS 2012
Friend groups within ego-nets

Use node features to find clusters:

[Albert, Einstein, German, Princeton]
Friend groups within ego-nets

Use node features to find clusters:

[Albert, Einstein, German, Princeton]

$p((x, y) \in E) \propto \exp \left\{ \sum_{C_k \supseteq \{x, y\}} \langle \phi(x, y), \theta_k \rangle - \sum_{C_k \not\supseteq \{x, y\}} \alpha_k \langle \phi(x, y), \theta_k \rangle \right\}$

- circles containing both nodes
- all other circles

friends under the same advisor
CS department friends
college friends
‘alters’ \( v_i \)
‘ego’ \( u \)

family members
highschool friends
Modeling with Cross-Associations

Summarize binary matrices by minimizing the number of bits to encode it.

Fully Automatic Cross-Associations
Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, Christos Faloutsos
KDD 2004
Modeling with Cross-Associations

Co-clustering of grant applications

Full Automatic Cross-Associations
Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, Christos Faloutsos
KDD 2004
Joint co-clustering

- **Cohesive clusters & anomalies**

**Given** adjacency matrix $A$ and feature matrix $F$

**Find** homogeneous blocks (clusters) in $A$ and $F$

---

**PICS:** Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs. Leman Akoglu, Hanghang Tong, Brendan Meeder, Christos Faloutsos. SDM 2012
Prediction with Co-clustering

ACCAMS: Additive Co-clustering to Approximate Matrices Succinctly
Alex Beutel, Amr Ahmed, Alex Smola
WWW 2015
Prediction with Co-clustering

ACCAMS: Additive Co-clustering to Approximate Matrices Succinctly
Alex Beutel, Amr Ahmed, Alex Smola
WWW 2015
Modeling with Co-clustering

ACCAMS: Additive Co-clustering to Approximate Matrices Succinctly
Alex Beutel, Amr Ahmed, Alex Smola
WWW 2015
1. Subgraph Analysis
   a) Background
   b) Normal Behavior
   c) Abnormal Behavior

2. Propagation Methods

3. Latent Factor Models
Fraud in Telecommunication Networks

- Community of Interest:
  - top-K connections

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- Community of Interest:
  - top-K connections

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- Community of Interest:  
  - top-K connections  
  - $d_2$ community includes the COI for neighbors

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- Community of Interest:
  - top-K connections
  - $d_2$ community includes the COI for neighbors
  - Label known fraudsters
  - Guilt-by-Association
    - If most nodes in your $d_2$ community are fraudulent, you are probably fraudulent.

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- Community of Interest:
  - top-K connections
  - $d_2$ community includes the COI for neighbors
  - Label known fraudsters
  - Guilt-by-Association
    - If most nodes in your $d_2$ community are fraudulent, you are probably fraudulent.

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- **Community of Interest:**
  - top-K connections
  - $d_2$ community includes the COI for neighbors
  - Label known fraudsters
  - **Guilt-by-Association**
    - If most nodes in your $d_2$ community are fraudulent, you are probably fraudulent.

**Communities of Interest**
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Fraud in Telecommunication Networks

- Community of Interest:
  - top-K connections
  - $d_2$ community includes the COI for neighbors
  - Label known fraudsters
- Guilt-by-Association
  - More “guilt-by-association” in next section

Communities of Interest
Corrinna Cortes, Daryl Pregibon, and Chris Volinsky
Springer, 2001
Pattern: Ego-net Power Law Density

Oddball: Spotting anomalies in weighted graphs
Leman Akoglu, Mary McGlohon, Christos Faloutsos
PAKDD 2010
Suspicious Subgraphs in Finance

Blackhole: Group of nodes with far more incoming weight than outgoing.

Could be indicative of trading ring buying up stock.
Suspicious Subgraphs in Finance

Volcano:
Group of nodes with far more outgoing weight than incoming.
Could be indicative of trading ring selling off inflated stock

Detecting Blackholes and Volcanoes in Directed Networks
Zhongmou Li, Hui Xiong, Yanchi Liu
ICDM 2010
Graph Cuts for Intrusion Detection

Bipartite graph between source IPs and destination IPs
Graph Cuts for Intrusion Detection

Connect source IPs if they connect to same destinations
Graph Cuts for Intrusion Detection

Nodes that cross communities are suspicious

Use min-cut to find graph cuts
## Practitioner’s Guide to Detecting Fraud

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>COI</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>OddBall</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Blackholes &amp; Volcanoes</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Anti)-Social</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SODA</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FocusCO</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gIceberg</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CopyCatch</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SynchoTrap</td>
<td>Bipartite+</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Co-Clustering</td>
<td>Bipartite*</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Outlier Detection in Attributed Subgraphs

User query: 3-author clique

Normal

Anomalous

Anomalous

Local Learning for Mining Outlier Subgraphs from Network Datasets
Manish Gupta, Arun Mallya, Subhro Roy, Jason Cho, Jiawei Han
SDM 2014 (slides adapted from Manish Gupta)
Outlier Detection in Attributed Subgraphs

User query: 3-author clique

Learn a Max-Margin SVM to predict which edges in the neighborhood exist based on node features.

Normal
Outlier Detection in Attributed Subgraphs

User query: 3-author clique

Learn a Max-Margin SVM to predict which edges in the neighborhood exist based on node features.

Local Learning for Mining Outlier Subgraphs from Network Datasets
Manish Gupta, Arun Mallya, Subhro Roy, Jason Cho, Jiawei Han
SDM 2014 (slides adapted from Manish Gupta)
Outlier Detection in Attributed Subgraphs

User query: 3-author clique

Learn a Max-Margin SVM to predict which edges in the neighborhood exist based on node features.

Normal

Local Learning for Mining Outlier Subgraphs from Network Datasets
Manish Gupta, Arun Mallya, Subhro Roy, Jason Cho, Jiawei Han
SDM 2014 (slides adapted from Manish Gupta)
Outlier Detection in Attributed Subgraphs

User query: 3-author clique

Learn a Max-Margin SVM to predict which edges in the neighborhood exist based on node features.

Normal

Local Learning for Mining Outlier Subgraphs from Network Datasets
Manish Gupta, Arun Mallya, Subhro Roy, Jason Cho, Jiawei Han
SDM 2014 (slides adapted from Manish Gupta)
Outlier Detection in Attributed Subgraphs

Graph G

Subgraph Query

Match 1  Match 2 … Match m

Outlier Score  Outlier Score  Outlier Score  Outlier Score  Outlier Score  Outlier Score

Top K

Local Learning for Mining Outlier Subgraphs from Network Datasets
Manish Gupta, Arun Mallya, Subhro Roy, Jason Cho, Jiawei Han
SDM 2014  (slides adapted from Manish Gupta)
Clustering and Outlier Detection in Attributed Graphs

Given a graph with node attributes,

Find focused clusters that are dense and share attributes, and

Detect outliers, nodes whose attributes deviate from their cluster’s attributes.
Clustering & Outlier Detection in Attributed Graphs

1. Examples
2. Inference "focus" attribute(s)
3. Detect focused clusters & outliers
4. Detect focused clusters & outliers

Age  Gender  Location
Clustering & Outlier Detection in Attributed Graphs

1. Clustering objective: conductance \( \phi^{(w)} \) weighted by focus

\[
\phi^{(w)}(C, G) = \frac{W_{cut}(C)}{WVol(C)}
\]

2. At each step in cluster expansion:
   2.1 - Examine boundary nodes
   2.2 - Add node with best \( \Delta \phi^{(w)} \)
   2.3 - Record best structural node

3. Focused Outliers: left-out best structural nodes
Focused Clustering and Outlier Detection in Large Attributed Graphs

Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sanchez, Emmanuel Muller

KDD 2014

(slides adapted from Bryan Perozzi)
A Probabilistic Approach to Uncovering Attributed Graph Anomalies
Nan Li, Huan Sun, Kyle Chipman,
Jemin George, Xifeng Yan
SDM 2014
Anomalous-Attribute Subgraphs

Crime Rates by State, 2008

GeoCurrents Map


- Infected
- Not Infected or Missing Data

Subgraph with skewed attribute distribution

A Probabilistic Approach to Uncovering Attributed Graph Anomalies
Nan Li, Huan Sun, Kyle Chipman, Jemin George, Xifeng Yan
SDM 2014
Anomalous-Attribute Subgraphs

Two generative processes:
1) anomaly distribution &
2) background distribution

Background: $V^{(0)}$  
Anomaly: $V^{(1)}$
Anomalous-Attribute Subgraphs

Two generative processes:
1) anomaly distribution &
2) background distribution

One overall mixture

\[ P(v_i) = \sum_{k=0}^{1} \theta_i^{(k)} P^{(k)}(v_i) \]

With probability \( \theta_i^{(0)} \), \( v_i \) belongs to the background component \( V^{(0)} \), and with \( \theta_i^{(1)} \) the anomaly component \( V^{(1)} \).
Anomalous-Attribute Subgraphs

Two generative processes:
1) anomaly distribution &
2) background distribution

One overall mixture

\[ P(v_i) = \sum_{k=0}^{1} \theta_i^{(k)} P^{(k)}(v_i) \]

With probability \( \theta_i^{(0)} \), \( v_i \) belongs to the background component \( V^{(0)} \), and with \( \theta_i^{(1)} \) the anomaly component \( V^{(1)} \).

Each component is a Bernoulli distribution

\[ P^{(k)}(v_i) = p^{(k)}(1) X_i (1 - p^{(k)}(1))^{1-X_i} \]
Anomalous-Attribute Subgraphs

Data loglikelihood of vertex set $V$

\[ \ell(V) = \sum_{v_i \in V} \log P(v_i) = \sum_{v_i \in V} \log \sum_k \theta_i^{(k)} P^{(k)}(v_i) \]
Anomalous-Attribute Subgraphs

Data loglikelihood of vertex set $V$
\[
\ell(V) = \sum_{v_i \in V} \log P(v_i) = \sum_{v_i \in V} \log \sum_k \theta_i^{(k)} P^{(k)}(v_i)
\]

Maximize:
\[
\ell(V) - \lambda R_N(\Theta) + \gamma R_E(\Theta)
\]

Network regularizer (enhances connectivity within each component)

Entropy regularizer (enhances polarity of mixture weights)

A Probabilistic Approach to Uncovering Attributed Graph Anomalies
Nan Li, Huan Sun, Kyle Chipman, Jemin George, Xifeng Yan
SDM 2014
glIceberg Anomalies

(a) Original Graph

(b) Vertices Arranged by Aggregate Score

Aggregate score: concentration of attribute in vertex’s vicinity

---

glIceberg: Towards Iceberg Analysis in Large Graphs
Nan Li, Ziyu Guan, Lijie Ren, Jian Wu, Jiawei Han, Xifeng Yan,
ICDE 2013
glIceberg: Towards Iceberg Analysis in Large Graphs

Nan Li, Ziyu Guan, Lijie Ren, Jian Wu, Jiawei Han, Xifeng Yan,
ICDE 2013
glIceberg Anomalies

Aggregation over PPVs
To compute $q$-scores

Forward aggregation

Backward aggregation

$p_u(v) = \frac{d_v}{d_u} p_v(u)$
To compute Aggregation of PPVs $q$-scores:

1. **Forward aggregation**
   - Starting from each black vertex, compute $q$-scores and contributions to other vertices.
   - Walks continue until $q$-scores saturate.

2. **Backward aggregation**
   - Starting from the entire vertex set, aggregate $q$-scores from each vertex.
   - Reduce $q$-scores from vertices until all contributions are accounted for.

**Thresholding**

- Define the threshold $\theta$ (e.g., $\theta = 0.5$).
- Filter vertices with $q$-scores above the threshold.

**Example**

- **Forward aggregation**
  - From vertex $a$, compute $q$-scores to $b$, $c$, $d$, etc.
  - Contribute to $b$, $c$, $d$, etc.

- **Backward aggregation**
  - From vertices $b$, $c$, $d$, etc., aggregate $q$-scores to $a$.

**Icierberg vertices**

- Vertices with $q$-scores above the threshold.

---

gIceberg: Towards Iceberg Analysis in Large Graphs
Nan Li, Ziyu Guan, Lijie Ren, Jian Wu, Jiawei Han, Xifeng Yan,
ICDE 2013
Practitioner’s Guide to Detecting Fraud

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>COI</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>OddBall</td>
<td>Undirected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blackholes &amp; Volcanoes</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Anti)-Social</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SODA</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FocusCO</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gIceberg</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CopyCatch</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SynchoTrap</td>
<td>Bipartite+</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Co-Clustering</td>
<td>Bipartite*</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Lockstep Behavior in the Graph

Dense group of data miner Page Likes

Dense group of purchased Page Likes

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

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

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

Find \([n,m,\Delta t,\rho]\)-Temporally Coherent Near Bipartite Cores (TNBC)
Lockstep Behavior in the Graph

CopyCatch works [quickly] – Few runs are enough

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

Temporal lockstep behavior found in Instagram followers

Uncovering Large Groups of Active Malicious Accounts in Online Social Networks
Qiang Cao, Xiaowei Yang, Jieqi Yu, Christopher Palow
ACM CCS 2014
Lockstep Behavior in the Graph

Accounts perform wide variety of synchronized tasks

Upload spammy photos
Share IP addresses (color)

Algorithmic Challenge: Repeated actions
Lockstep Behavior in the Graph

SynchoTrap

- Define edge weight by similarity of actions (including time, IP, action, etc.)
- Cluster to find synchronized users
Uncovering Large Groups of Active Malicious Accounts in Online Social Networks

Qiang Cao, Xiaowei Yang, Jieqi Yu, Christopher Palow
ACM CCS 2014
Co-clustering to find network fraud

Handles binary features (edges without side information) e.g., connection type

Connections

- HTTP
- DNS
- rate
- su_failed
- count
Co-clustering to find network fraud

As well as features with continuous values (edges with side information) e.g., round-trip time, number of requests, etc.
Co-clustering to find network fraud

Co-clustering finds groups of connections with very similar edges through partitioning all rows and columns.

Connections
- HTTP
- DNS
- rate
- su_failed
- count

Network Anomaly Detection using Co-Clustering
Evangelos Papalexakis, Alex Beutel, Peter Steenkiste
ASONAM 2012
Co-clustering to find network fraud

Each cluster is nearly all normal connections or all attacks

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Number of Connections</th>
<th>Percent Normal</th>
<th>Percent Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20,156</td>
<td>97.74%</td>
<td>2.26%</td>
</tr>
<tr>
<td>2</td>
<td>116,822</td>
<td>5.30%</td>
<td>94.70%</td>
</tr>
<tr>
<td>3</td>
<td>29,591</td>
<td>93.34%</td>
<td>6.66%</td>
</tr>
<tr>
<td>4</td>
<td>281,437</td>
<td>0.21%</td>
<td>99.79%</td>
</tr>
<tr>
<td>5</td>
<td>46,014</td>
<td>93.85%</td>
<td>6.15%</td>
</tr>
</tbody>
</table>

Network Anomaly Detection using Co-Clustering
Evangelos Papalexakis, Alex Beutel, Peter Steenkiste
ASONAM 2012
## Practitioner’s Guide to Detecting Fraud

<table>
<thead>
<tr>
<th>Method</th>
<th>Graph Type</th>
<th>Node Attributes</th>
<th>Edge Attributes</th>
<th>Seed Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>COI</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>OddBall</td>
<td>Undirected</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Blackholes &amp; Volcanoes</td>
<td>Directed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Anti)-Social</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SODA</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FocusCO</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gIceberg</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CopyCatch</td>
<td>Bipartite</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>SynchoTrap</td>
<td>Bipartite+</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Co-Clustering</td>
<td>Bipartite*</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>PICS</td>
<td>Undirected</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recap

- **COI:** Guilt-by-Association
- **Oddball:** Unusually dense graphs are suspicious (along with other surprising patterns described in the paper)
- **Blackholes and Volcanos** can be indicative of trading rings
- **(Anti)social behavior** – In packet traces, cliques are normal and bridges connecting cliques are suspicious
- **SODA:** Attributed subnetwork anomalies
- **FocusCO:** Learn model of normal attributes among communities and find outliers in the community
- **gIceberg:** Subgraph with anomalous distribution of attribute
- **CopyCatch:** Temporally near-bipartite cores are extra-suspicious
- **SynchoTrap:** Generalize CopyCatch to handle extra data like IP addresses and repeat actions
- **Co-clustering:** Global partitioning to find locally similar regions; can include edges with side information.