Anomaly Detection - Graphs

Xin Luna Dong, Christos Faloutsos

Xian Li, Subhabrata Mukherjee, Prashant Shiralkar

KDD 2018
Trust <-> Anomalies <-> Patterns - example1
Trust <-> Anomalies <-> Patterns - example 2

Same 300 people, re-tweeting the same 500 messages
Trust <-> Anomalies <-> Patterns
Roadmap

- Single-node anomalies -> ‘oddball’ ++
- Group anomalies -> ‘copyCatch’ ++
Single-node anomalies - Problem sketch
Single-node anomalies - Problem sketch

Leman Akoglu, Mary McGlohon, Christos Faloutsos: 
OddBall: approach

1) from each node, extract ‘ego-net’
   1.1) extract features
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 principle eigenvalue

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

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

\[ \lambda_{w,i} \propto \sqrt{W} \]

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

\[ \lambda_{w,i} = W \]

\[ \lambda_{w,i} \approx W \]

N: #neighbors, W: total weight
OddBall: pattern#1

discussion group, “rank boosting”, etc.

telemarketer, spammer, port scanner, “popularity contests”, etc.

slope=2
slope=1.35
slope=1
OddBall: pattern #2

high $ vs. #accounts, high $ vs. #donors, etc.

slope = 1.08

slope = 1

uniform, robot-like behavior

#edges E

total weight W

KDD 2018
OddBall: pattern#3

\[ \text{total weight } W \]

\[ \lambda_{1,w} \]

- slope = 1
- slope = 0.64
- slope = 0.5
OddBall: anomaly detection

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

\[
\text{score}_{\text{dist}} = \text{distance to fitting line} \\
\text{score}_{\text{outl}} = \text{outlier-ness score} \\
\text{score} = \text{func}(\text{score}_{\text{dist}}, \text{score}_{\text{outl}})
\]
# OddBall: datasets

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

**Unipartite** graphs:  
4. BlogNet  
   - |V|: 27K  
   - |E|: 126K  
5. PostNet  
   - |V|: 223K  
   - |E|: 217K  
6. Enron  
   - |V|: 36K  
   - |E|: 183K  
7. AS peering  
   - |V|: 11K  
   - |E|: 8K
OddBall at work (Posts)

223K posts
217K citations

http://www.sizemore.co.uk/2005/08/l-feel-some-movies-coming-on.html

#citations
#cross-citations
OddBall at work (FEC)

COM2CANDIDATES

Kerry, John F.

Snyder, James E. Jr

Russo, Aaron

6K candidates 125K checks
OddBall at work (DBLP)
Conclusions - Anomaly detection in graphs

1) Single-node: OddBall (and many more...)
NUMEROUS extensions: #1- node-attributes

Nodes have attributes (age, gender, \$income, ...)

- Leman Akoglu, Hanghang Tong, Brendan Meeder, Christos Faloutsos: PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs. SDM 2012
NUMEROUS extensions: #2 – time evolving

Time-evolving graphs (who-calls-whom-when)

- Evangelos E. Papalexakis, Christos Faloutsos, Tom M. Mitchell, Partha Pratim Talukdar, Nicholas D. Sidiropoulos, Brian Murphy: *Turbo-SMT: Accelerating Coupled Sparse Matrix-Tensor Factorizations by 200x*. SDM 2014:

- Miguel Ramos de Araujo, Pedro Manuel Pinto Ribeiro, and Christos Faloutsos, *TensorCast: Forecasting with Context using Coupled Tensors*, IEEE ICDM 2017 (Best Paper Award)
NUMEROUS extensions: #3 – w/ labels

Some labels (‘fraud’/‘honest’), exist

Roadmap

- Single-node anomalies -> ‘oddball’
- Group anomalies -> ‘copyCatch’ ++
Fraud

Given

- Who ‘likes’ what page, and when

Find

- Suspicious users and suspicious products

Fraud

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

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
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

Manually labeled 22 randomly selected clusters from February 2013

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

Fake acct

Fake Accounts
Malicious Browser Extensions
OS Malware
Credential Stealing
Social Engineering

58%
23%
9%
5%
5%

KDD 2018
Roadmap

- Single-node anomalies -> ‘oddball’
- Group anomalies -> ‘copyCatch’ ++
  - Stealthy attackers?
Problem: Social Network Link Fraud

Target: find “stealthy” attackers missed by other algorithms

Bipartite core

Clique

41.7M nodes
1.5B edges
Problem: Social Network Link Fraud
Target: find “stealthy” attackers missed by other algorithms

Lekan Olawole Lowe @loweinc 26 Jul 09
Sign up free and Get 400 followers a day using http://tweeteradder.com

Lekan Olawole Lowe @loweinc 26 Jul 09
Get 400 followers a day using http://www.tweeterfollow.com

Takeaway: use reconstruction error between true/latent representation!

Numerous extensions - #1: SVD-based

- B. Aditya Prakash, Ashwin Sridharan, Mukund Seshadri, Sridhar Machiraju, Christos Faloutsos: *EigenSpokes: Surprising Patterns and Scalable Community Chipping in Large Graphs*. PAKDD (2) 2010: 435-448

- Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos, Shiqiang Yang: *Inferring Strange Behavior from Connectivity Pattern in Social Networks*. PAKDD (1) 2014: 126-138
Numerous extensions - #2: dense-blocks

Camouflage

- Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, Christos Faloutsos: *FRAUDAR: Bounding Graph Fraud in the Face of Camouflage*. KDD 2016: 895-904
Conclusions - graph anomaly detection

- Single-node anomalies -> ‘oddball’ (PICS, ++)
- Group anomalies -> ‘copyCatch’ (SVD, Fraudar, ++)