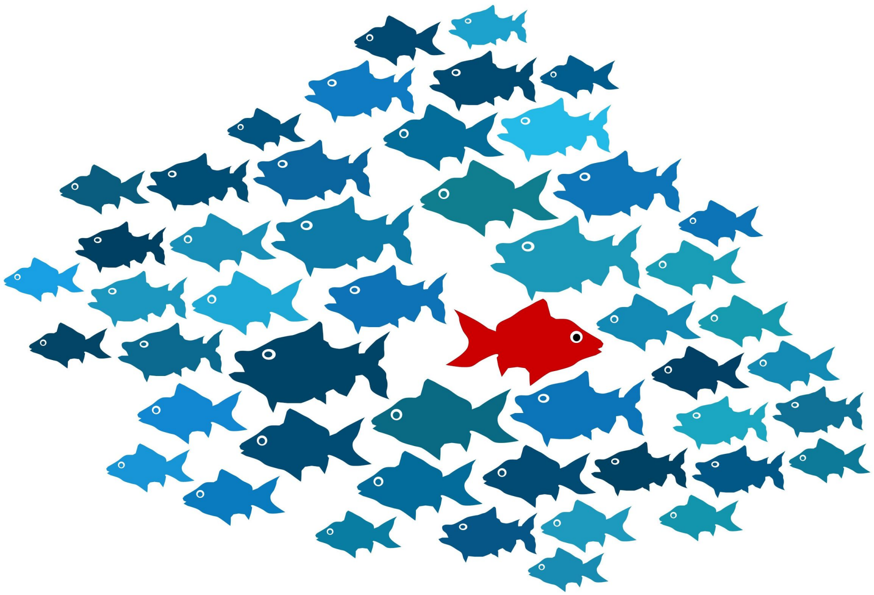
Pattern Detection to Identify Anomalous Users

Team R-Clique
Daniel Geng, Meghana Ginjpalli, Sara Melvin,
Sonu Mishra, and Andrew Wong

Outline

- Background
- State-of-the-art Models
  - Temporal-based Approach: RSC
  - Group-based Approach: ND-SYNC
  - Group-based Approach: GLAD
  - Graph-based Approach: EdgeCentric
  - Bayesian-based Approach: BIRDNEST
- Summary



Background

Outline

- Background
- State-of-the-art Models
  - **Temporal-based Approach: RSC [1]**
  - Synchronicity Approach: ND-SYNC
  - Group-based Approach: GLAD
  - Graph-based Approach: EdgeCentric
  - Bayesian-based Approach: BIRDNEST
- Summary

Rest-Sleep-and-Comment (RSC)

Purpose: Determine if a user is a human or a bot in social media

Solution: Temporal approach

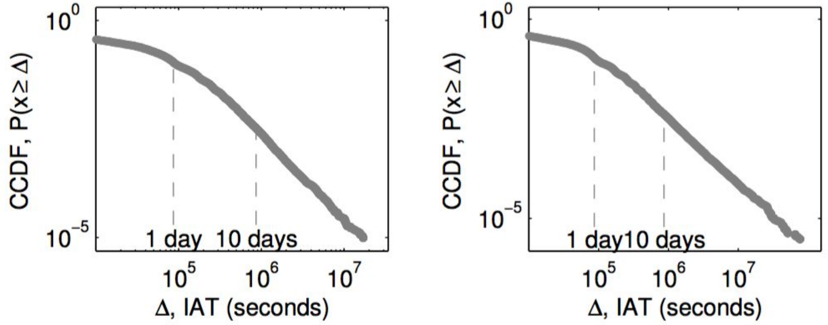
Builds on ideas from previous works (CNPP, Poisson Process, SFP)

- Identify patterns using only posting inter-arrival time (IAT) distributions
- Use real user patterns to build a model
- Use model to classify anomalies

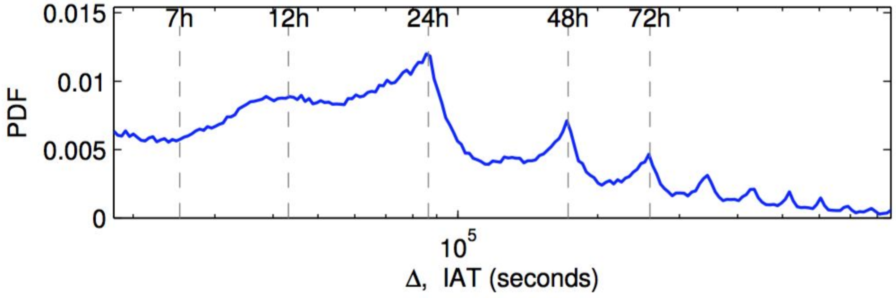
RSC - Observed IAT Patterns

Observation 1: Heavy-tailed distribution: indication that human users can be inactive for long periods of time before making a post

Observation 2: Periodic spikes: effect of circadian rhythm on posting times in every 24 hour period



(a) Reddit (b) Twitter

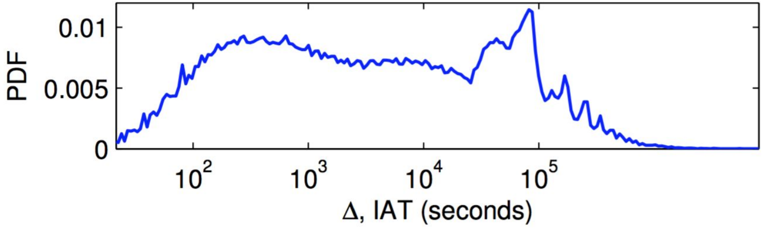


(a) Reddit Dataset

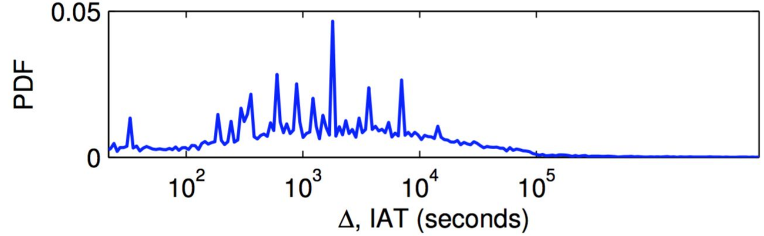
RSC - Observed IAT Patterns

Observation 3: Bimodal distribution: two “humps” in IAT distribution

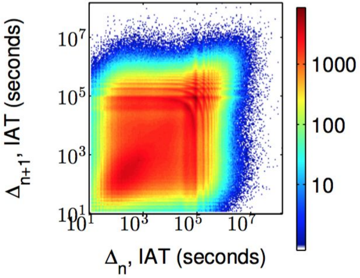
Observation 4: Positive correlation: the IAT between two postings is dependent on the previous IAT



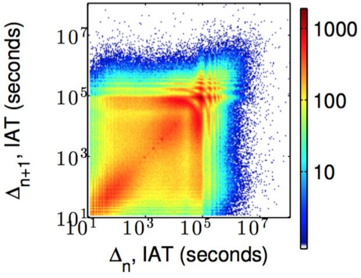
(a) Humans



(b) Bots



(a) Reddit



(b) Twitter

RSC - Algorithm

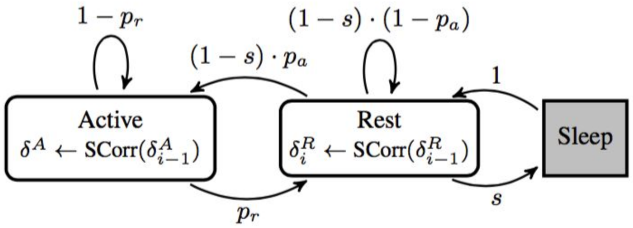
1) Self-Correlated Process (SCorr) - addresses observation 1 (Heavy tail) and observation 4 (Positive Correlation)

DEFINITION 3. *Let δ_i be the inter-arrival time between the events i and $i-1$. A stochastic process is a Self-Correlated Process, with base rate λ and correlation ρ if:*

$$\boxed{\begin{aligned} \delta_1 &\sim Exp\left(\frac{1}{\lambda}\right) \\ \delta_i &\sim Exp\left(\rho \cdot \delta_{i-1} + \frac{1}{\lambda}\right) \end{aligned}} \qquad \begin{aligned} &(1) \\ &(2) \end{aligned}$$

where $X \sim Exp(1/\lambda)$ denotes an exponentially distributed random variable with rate λ.

2) Active, rest, and sleep states - addresses observations 2 (Periodic spiking) and observation 3 (Bimodal)

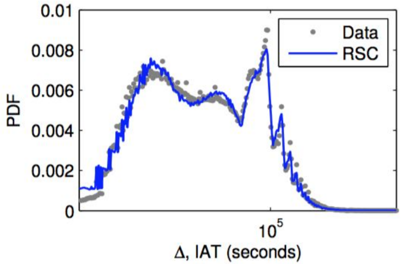


$$s = \begin{cases} 1, \text{ if } t_{\text{wake}} < t_{\text{clock}} < t_{\text{sleep}}, \\ 0, \text{ otherwise.} \end{cases}$$

RSC - Algorithm

3) Estimate RSC Parameters using timestamps from all users

4) Bot Detection (RSC-Spotter)

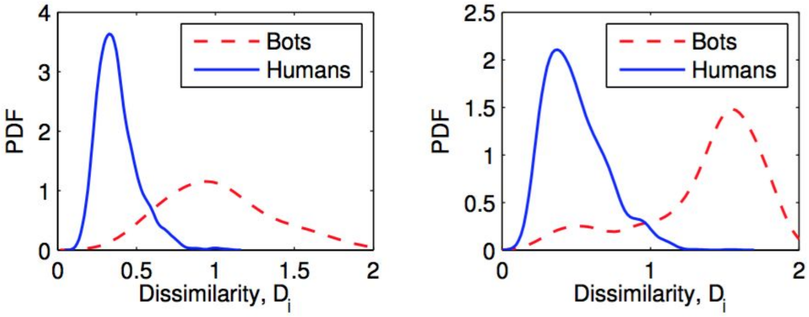


(a) **RSC**

- For each user:
- 1) Compute the IAT histogram
- 2) Generate synthetic timestamps with RSC
- 3) Compare user and synthetic IAT histogram to decide if a user is a bot given the dissimilarity value (sum of squared differences)

RSC-Spotter Evaluation

- 2,000 reddit users (1,963 humans, 37 bots)
- 1,353 Twitter users (1,289 verified users, 64 bots)
- Dataset was randomly split into same sized training/test subsets with the same class distribution



(a) Reddit (b) Twitter

Kernel smoothing function estimate of dissimilarity values

Pros and Cons of RSC

Pros

- >94% bot detection precision
- does not assume events to be independent and identically distributed (i.i.d)
- covers more communication patterns compared to other approaches

Cons

- only uses one feature
- models a time pattern to detect outliers that may not be useful to other applications

Outline

- Background
- State-of-the-art Models
  - Temporal-based Approach: RSC
  - **Synchronicity Approach: ND-SYNC [2]**
  - Group-based Approach: GLAD
  - Graph-based Approach: EdgeCentric
  - Bayesian-based Approach: BIRDNEST
- Summary

ND-SYNC

- Problem: detect suspicious users based on their retweet threads with large numbers of retweets (100+)
- Use similarity of user’s threads and everyone else’s threads across many features to detect anomalies
- Approach:
  - Feature subspace sweeping
  - User scoring based on suspiciousness score
  - Multivariate outlier detection for suspicious users

ND-SYNC Features

- Number of retweets
- Response time of first retweet
- Lifespan, constrained to three weeks
- RT-Q3 response time, time to garner the first ¾ of the retweets
- RT-Q2 response time, time to garner the first ½ of the retweets
- Arr-MAD, mean absolute deviation of interarrival times
- Arr-IQR, inter-quartile range of interarrival times

ND-SYNC Definitions

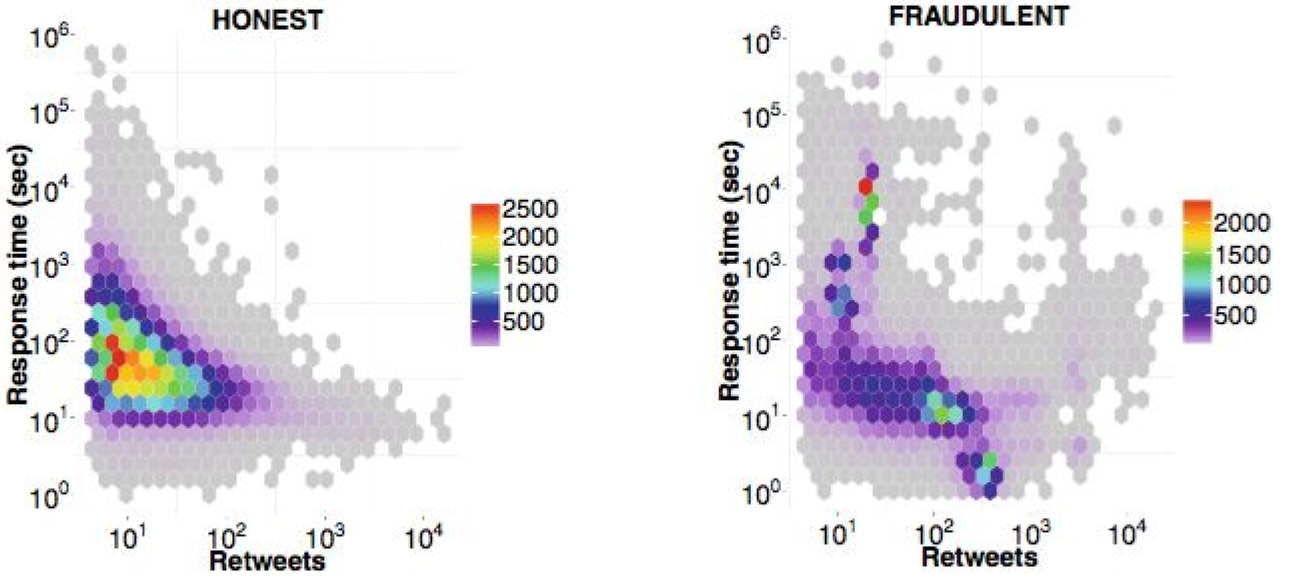
Synchronicity (intra-synchronicity): for a group of user's retweets, the average closeness between all pairs of its members

Normality (inter-synchronicity): for a group G' with respect to super-group G, the average closeness of members of G' to G

Residual score: difference between synchronicity of G' and lower bound on intra-synchronicity, defined with respect to Gs normality

***Suspiciousness**:* residual score of a projection of a group onto subspace; intuitively standardized measure of synchronicity (with respect to subspace)

ND-SYNC Data Visualization



[2] Giatsoglou, M. et. al. “ND-Sync: Detecting Synchronized Fraud Activities.” Advances in Knowledge Discovery and Data Mining, 2015, 201-214

ND-SYNC Algorithm

Generate all 2^p subspaces F and project data onto each

```
for each user's retweet threads t:
        for each subspace f in F:
                calculate suspiciousness of t in f
        put suspiciousness values over all subspaces into vector

repeatedly:
        extract outliers in suspiciousness with ROBPCA-AO

find set of users S by voting on outliers
```

Pros and Cons of ND-SYNC

Pros

- High predictive accuracy
  - 95%-97% accuracy
- Plotting features against each other yields intuitive explanations

Cons

- Intuitions from plotting are explainable, but unused in design of the model
- Model is large
- Classifies all users at once

Outline

- Background
- State-of-the-art Models
  - Temporal-based Approach: RSC
  - Synchronicity Approach: ND-SYNC
  - **Group-based Approach: GLAD [3]**
  - Graph-based Approach: EdgeCentric
  - Bayesian-based Approach: BIRDNEST
- Summary

GLAD

- Goal: determine group anomalies using GLAD (**G**roup **L**atent **A**nomaly **D**etection)
- Input: pointwise and pairwise data
- Algorithm determines groups and group anomalies simultaneously
- Observations:
    - social media contains both pointwise and pairwise data
    - group anomaly is harder to detect compared to individual anomaly
    - groups can be dynamic which makes it difficult to detect group anomalies

GLAD - Background

- What is a **group**?
  - Mixture model of a user behavior mixture model
- What is the **role mixture rate**?
  - An inference of group membership and role identity of each individual in a group
- What is a **group anomaly**?
  - occurs when role mixture rate is significantly different than a normal group role mixture rate

GLAD - Algorithm

- For each individual:
  - identify membership distribution using Dirichlet prior
  - calculate likelihood of an individual belonging to that group
  - For all other individuals:
    - Use Bernoulli distribution to check if a link is formed which occurs when two individuals have the same group identity
  - calculate role of individual by taking likelihood given role mixture rate and group identity
- **NOTE**: each individual can have multiple roles and can be in multiple groups

DGLAD - Algorithm

- For each timestep:
  - Apply GLAD algorithm
  - For each group:
    - calculate role mixture rate using Gaussian distribution on previous role mixture rate
- **NOTE**: If role mixture rate significantly changes over time, then a group anomaly is detected

Pros and Cons of GLAD/DGLAD

Pros

- Can identify groups and anomalies at the same time
- DGLAD can detect group anomalies in dynamic data

Cons

- The authors inserted group anomalies into the data
- DGLAD is computationally expensive and is not scalable

Outline

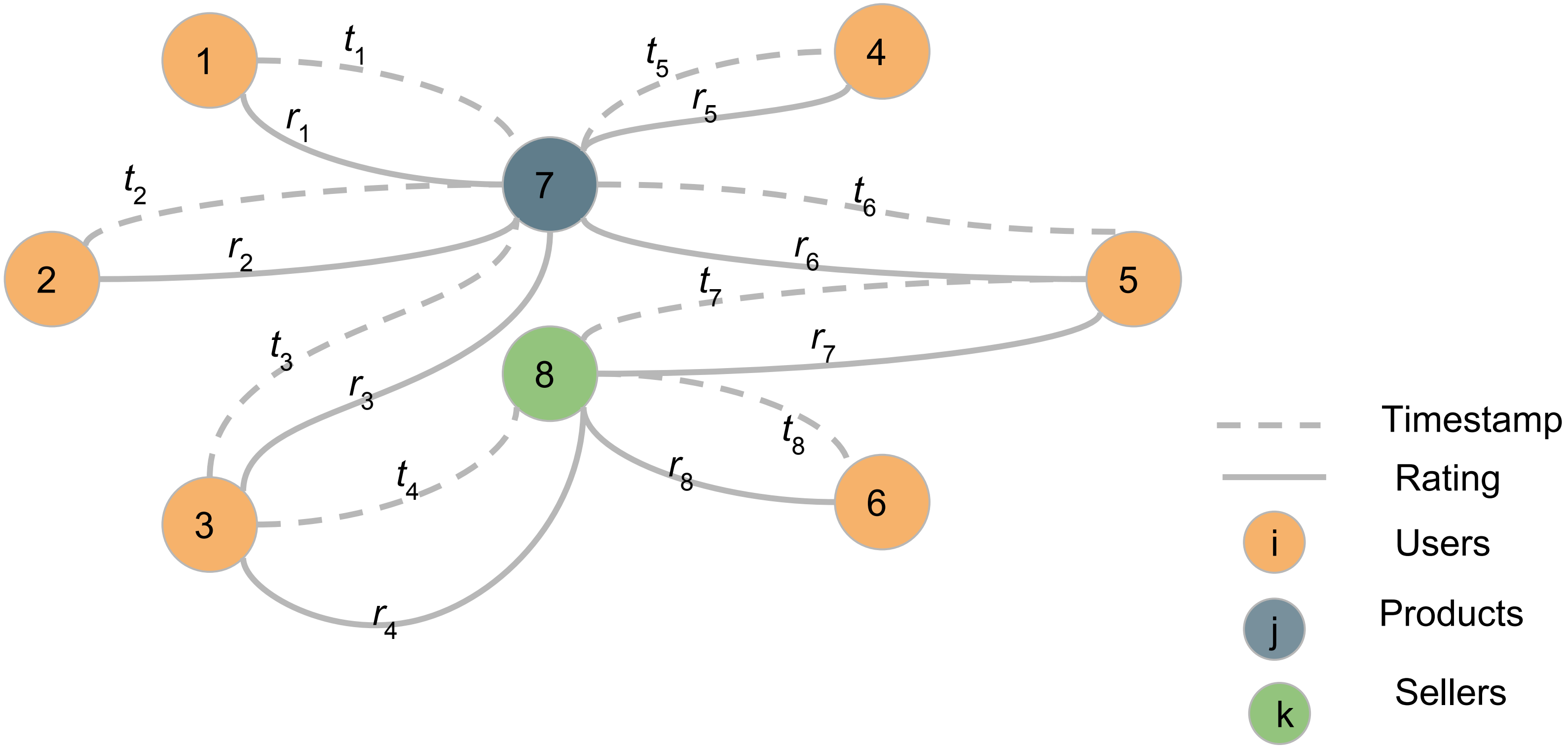
- Background
- State-of-the-art Models
  - Temporal-based Approach: RSC
  - Group-based Approach: ND-SYNC
  - Group-based Approach: GLAD
  - **Graph-based Approach: EdgeCentric [4]**
  - Bayesian-based Approach: BIRDNEST
- Summary

EdgeCentric

Goal: Detect anomalies in Edge-Attributed Networks

- Edge-Attributed graphs:
  - e-Commerce
  - Social Networks
  - Phonecall Networks
- Attributed edges describe how the adjacent nodes interact
- Edge-weighted graphs ?

EdgeCentric



EdgeCentric

Formulation:

- Given: $G\ (V, E, m)$, a static graph with numerical/categorical edge-attributes
- To devise: δ(.), an abnormality function to score each node
- Objective: identify the most irregular nodes in a scalable fashion

EdgeCentric

Formulation: Base Case

- Assumptions:
  - Single model-distribution, *C*
  - Single edge-attribute
  - Single relation
- $\delta_{base}(v) = |f_v| . \mathrm{KL}(\mathcal{V} \| C)$ where,
  - $|f_v|$: cardinality of edge-attribute value vector
  - $\mathcal{V}$: discrete probability distribution over chosen attribute
  - $\mathrm{KL}(\mathcal{V} \| C)$: KL divergence capturing the difference between $\mathcal{V}$ and C

EdgeCentric

Formulation: Multifaceted

- Assumptions:
  - ~~Single model-distribution, C,~~
  - Single edge-attribute
  - Single relation
- $\delta_{mf}(v) = |f_v| . \sum_g \left(\rho_{b,g} \, \mathrm{KL}(\mathscr{v} \, || C_{b,g}) \right)$ where,
  - $C_{b,g}$: gth model distribution of type *b* node
  - $\rho_{b,g}$: proportion of g^{th} cluster

EdgeCentric

Formulation: Multifaceted and Multi-attribute

- Assumptions:
  - ~~Single model-distribution, C,~~
  - ~~Single edge-attribute~~
  - Single relation
- $\delta_{ma}(v) = |f_v|. \sum_w \left(\sum_g \left(\rho_{b,w,g} \, \mathrm{KL}(\mathcal{v}_w \,||C_{b,w,g})\right)\right)$ where,
  - $C_{b,w,g}$: gth model distribution of type b node on w^{th} attribute
  - $\rho_{b,w,g}$: proportion of g^{th} cluster on w^{th} attribute

EdgeCentric

Formulation: Multifaceted and Multi-attribute

- Assumptions:
  - ~~Single model distribution, C,~~
  - ~~Single edge-attribute~~
  - ~~Single relation~~
- $\delta(v) = \sum_r \left(|f_{v,r}| \sum_w \left(\sum_g \left(\rho_{b,r,w,g} \, \mathrm{KL}(v_w \,||\, C_{b,r,w,g}) \right) \right) \right)$ where,
  - $C_{b,w,g}$: gth model distribution of type b node on w^{th} attribute and r^{th} relation
  - $\rho_{b,w,g}$: proportion of g^{th} cluster on w^{th} attribute and r^{th} relation
- $\delta(v)$: intuitively the expected # extra bits required to encode the node v's edge-attribute vectors w.r.t. a joint model over multiple relations, attributes and clusters

EdgeCentric

Methodology: Proposed EdgeCentric

Algorithm 1 EDGECENTRIC

Input: graph G
Output: sorted abnormality score vector for each node type in G

1: For each node in G, aggregate attribute values from outgoing edges per-relation-type.
2: Based on attribute type and range of values, discretize the space categorically for categorical attributes, and linearly or logarithmically for numerical attributes. Bin the per-node aggregated attribute values accordingly and normalize to construct probability mass functions.
3: For each node-type and attribute, cluster the vectors describing the per-attribute probability mass functions associated with each relation.
4: For each node-type, compute the abnormality score δ for all nodes over associated relations and attribute clusters.
5: For each node-type, sort (descending) the resulting abnormality scores and return with node indices.

Pros and Cons of EdgeCentric

Pros

- Leveraging multiple relationships between entities
- Can be applied to numerous applications

Cons

- Different attributes, relations, and cluster distributions are assumed to be independent.

Outline

- Background
- State-of-the-art Models
  - Temporal-based Approach: RSC
  - Group-based Approach: ND-SYNC
  - Group-based Approach: GLAD
  - Graph-based Approach: EdgeCentric
  - **Bayesian-based Approach: BIRDNEST [5]**
- Summary

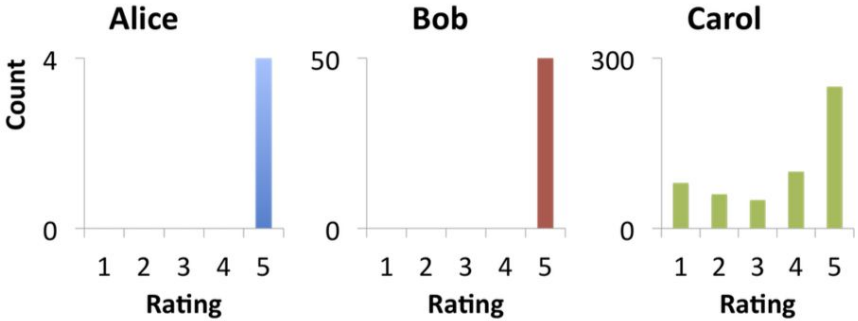
BIRDNEST

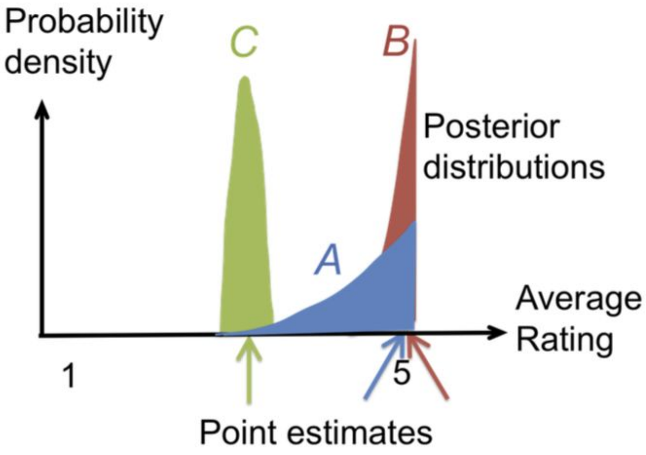
- BIRDNEST = **B**ayesian **I**nference for **R**atings-fraud **D**etection **N**ormalized **E**xpected **S**urprise **T**otal
- Goal
    - Given temporal rating data by users on products, BIRDNEST attempts to **compute how suspicious** a user is with a score
- Features Considered:
    - Inter-arrival Time (IAT)
    - Ratings
- Two part approach:
    - BIRD - building the Bayesian mixture model
    - NEST - computing suspiciousness of a user

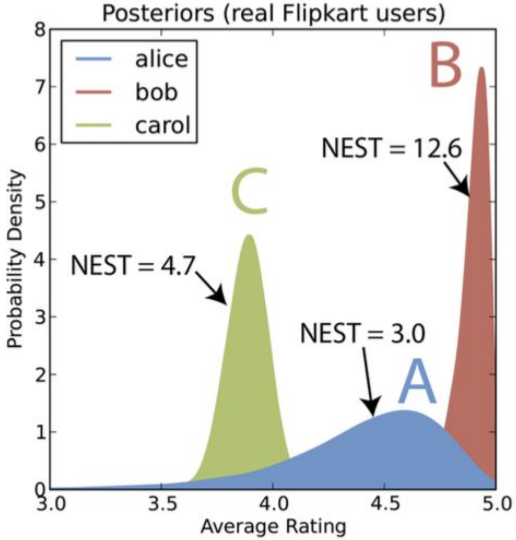
BIRDNEST - Intuition

Two typical questions a person would ask when determining anomalous behavior:

1) What is the distribution of a user? - BIRD
2) How suspicious is that distribution? - NEST







Figures from [5]

Pros and Cons of BIRDNEST

Pros

- Determines how anomalous instead of just classifying
- Can capture bot behavior if only one of the characteristics is present
  - 100% precision for top 50 suspicious users (Flipkart data)
- Fast

Cons

- Assumes independence of each rating event

Summary - What's the Best Algorithm?

Temporal-based Approach - Rest-Sleep-Comment (RSC)

Features with regular temporal patterns (e.g. most social networks)

Group-based Approach - GLAD, ND-SYNC

Situations where users share certain properties (e.g. Twitter)

Bayesian-based Approach - BIRDNEST

Situation where crowdsourcing of user rating is important (e.g. Yelp)

Graph-based Approach - EdgeCentric

Takes into account multiple relationships such as "user rates product," "user rates seller," etc. (e.g. eBay, Etsy)

References

1. Costa, A.F. et. al. “RSC: Mining and Modeling Temporal Activity in Social Media.” Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 269-278
2. Giatsoglou, M. et. al. “ND-Sync: Detecting Synchronized Fraud Activities.” Advances in Knowledge Discovery and Data Mining, 2015, 201-214
3. R. Yu, X. He, Y. Liu. “GLAD: Group Anomaly Detection in Social Media Analysis.” ACM. August 2014.
4. Shah, N. et. al. “EdgeCentric: Anomaly Detection in Edge-Attributed Networks.” arXiv:1510.05544.
5. Hooi, B., N. et. al. “BIRDNEST: Bayesian Inference for Ratings-Fraud Detection.” arXiv:1511.06030. Nov 2015.

Questions?