Outlier Detection for Mining Social Misbehavior

Neil Shah
Snap Research

08/20/2018

Snap Inc.
Carnegie Mellon University

About me

- Research Scientist at Snap (previously CMU)
- Interested in data mining, security, user-behavior modeling and network science
- Broadly focus on characterizing, detecting and mitigating online social misbehavior

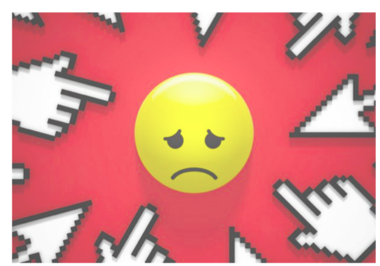
http://www.cs.cmu.edu/~neilshah/

Snap Inc. Carnegie Mellon University

What is social misbehavior?

Malicious behavior on social platforms which is unintended by creators or harmful to users

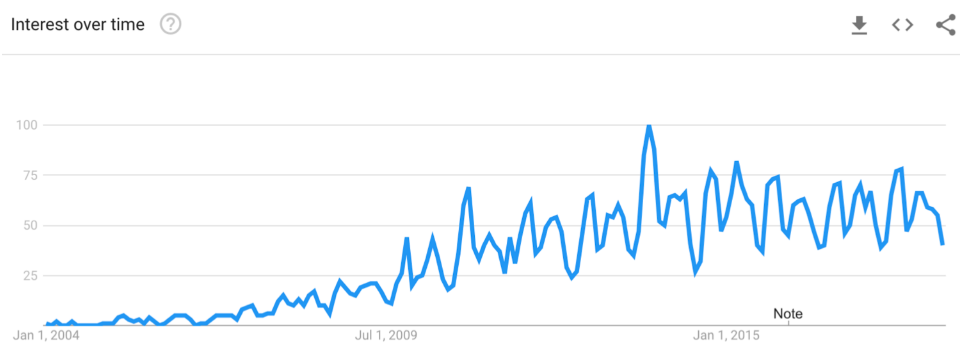
- Impacts user perception (spam, false information)
- Impacts user safety (malicious URLs, account compromise, blackmail, bullying)



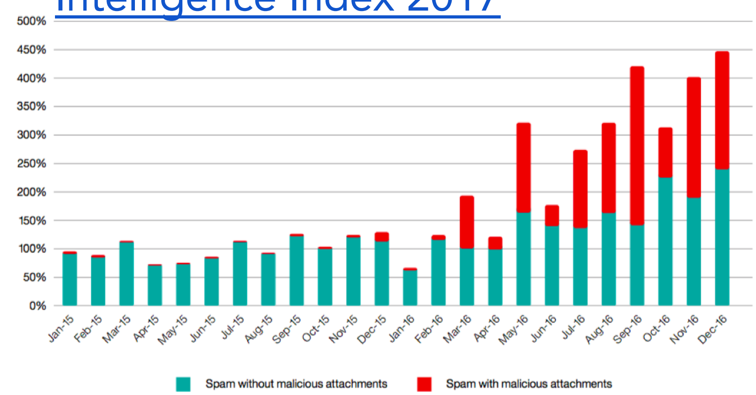
Social misbehavior is on the rise

- ~13-15% fake and duplicate accounts on Facebook/Twitter respectively[1,2]
- ¼ Americans visited a misinformative website around the 2016 election[3]

Increased interest in cyberbullying – [Google Trends]



Growth in email spam volume and bad attachments – [IBM Threat Intelligence Index 2017]

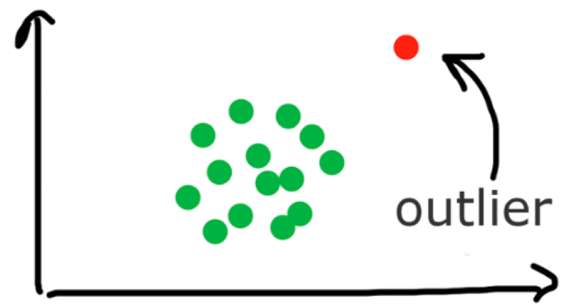


[1] Selective Exposure to Misinformation
[2] Facebook Q3'17 Earnings Report
[3] Online Human-bot Interactions: Detection, Estimation and Characterization

Outlier detection to the rescue

- Most generally, outlier detection is about finding *unlikely samples* in data

> *An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism (Hawkins, 1980).*



- In social settings, our samples are often *users*
- **We can tackle a wide variety of misbehavior detection tasks by identifying the right types of outlying users.**

Two examples

- Spotting suspicious link behavior in online social networks

- Combating fake viewership on livestreaming platforms

Two examples

- **Spotting suspicious link behavior in online social networks**

- Combating fake viewership on livestreaming platforms

Popularity on social media

- Measured inherently by numbers; on social networks, followers are the target metric

Following 620K

Followers 102M

Following 53

Followers 1.87M

Carnegie Mellon University

Following 910

Followers 48.5K

Following 10

Followers 14.7M

Gamifying popularity

When a measure becomes a target, it ceases to be a good measure. (Goodhart, 1975)

Report: 92% of Newt Gingrich's Twitter Followers Aren't Real

As many as 48 million Twitter accounts aren't people, says study

Buy Twitter Followers with Quick Delivery

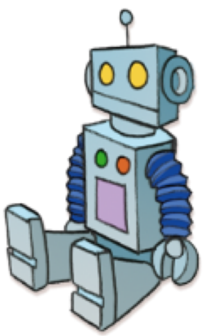
Socialshop offers the best Twitter followers in the market. Check out our deals!

Micro	Mini	Starter	Standard	Medium	Premium
$2 One Time Fee	$5 One Time Fee	$6 One Time Fee	$13 One Time Fee	$22 One Time Fee	$40 One Time Fee
100 Followers	500 Followers	1000 Followers	2500 Followers	5000 Followers	10.000 Followers
High Quality	High Quality	High Quality	High Quality	High Quality	High Quality
100% Safe	100% Safe	100% Safe	100% Safe	100% Safe	100% Safe
E-mail Support	E-mail Support	E-mail Support	E-mail Support	E-mail Support	E-mail Support
Super fast delivery	Super fast delivery	Super fast delivery	Super fast delivery	Super fast delivery	Super fast delivery
Buy Now	Buy Now	Buy Now	Buy Now	Buy Now	Buy Now

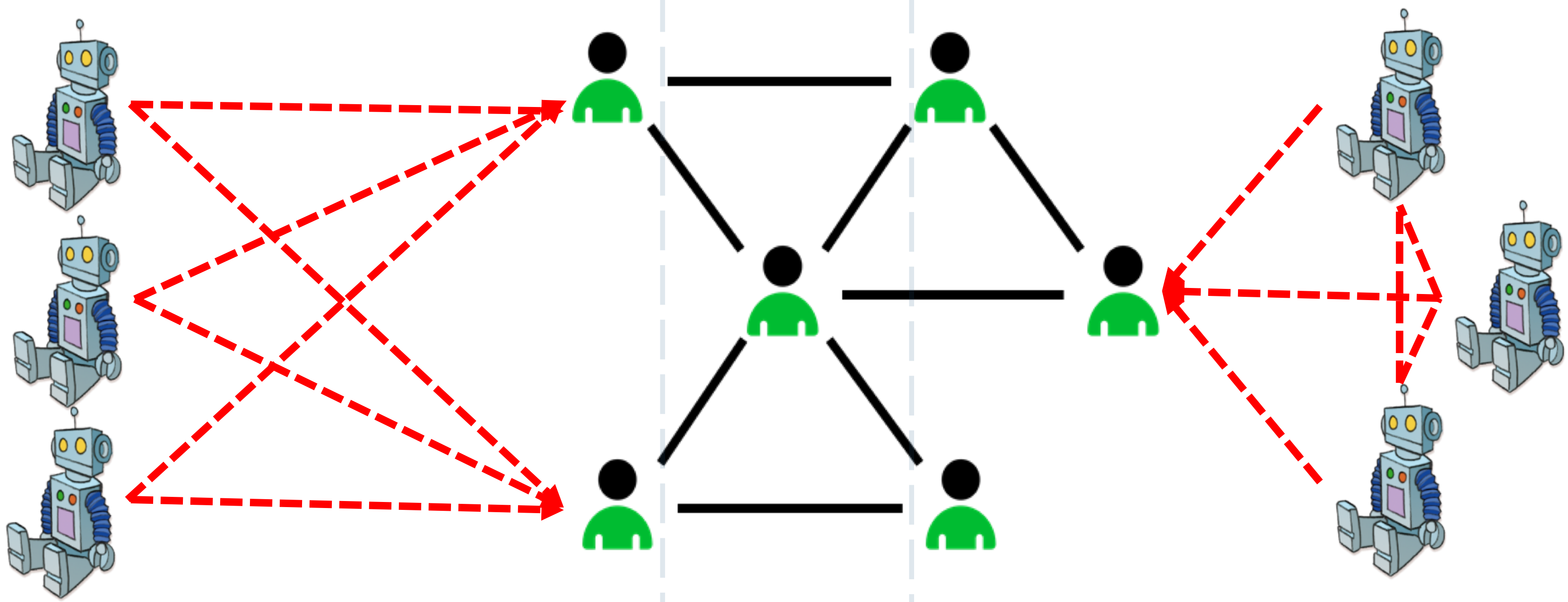
Problem definition

Given: *a static, social graph G*
Find: *nodes which are fake followers ("link fraud")*

- Ubiquitous problem in social media
- Disruptive to recommendation
- Harmful to user trust



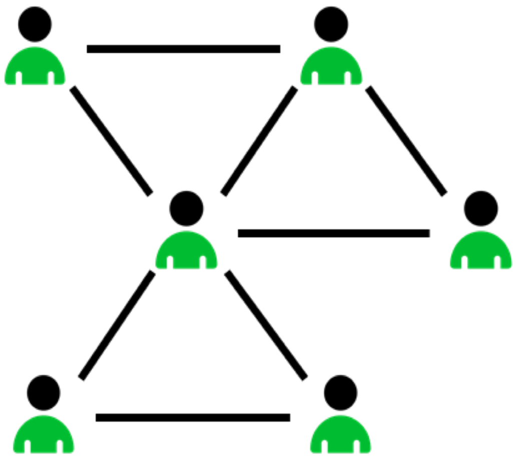
Link fraud visualized



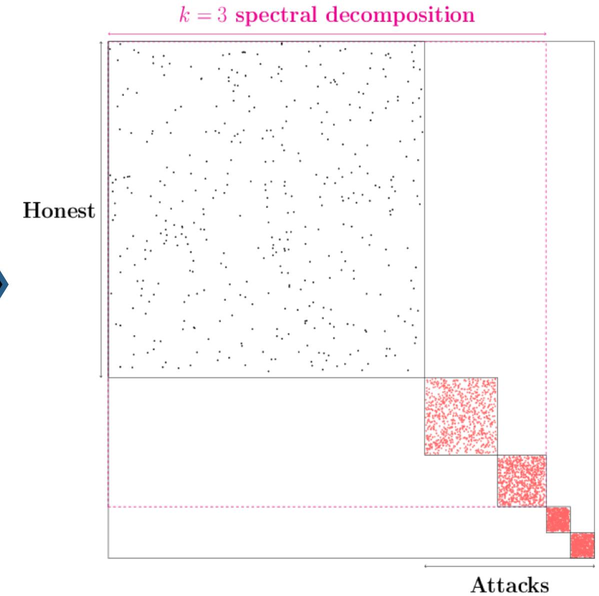
Bipartite core: nodes used only to follow customers

Clique: nodes follow each other, but also some customers

Decomposition for detection



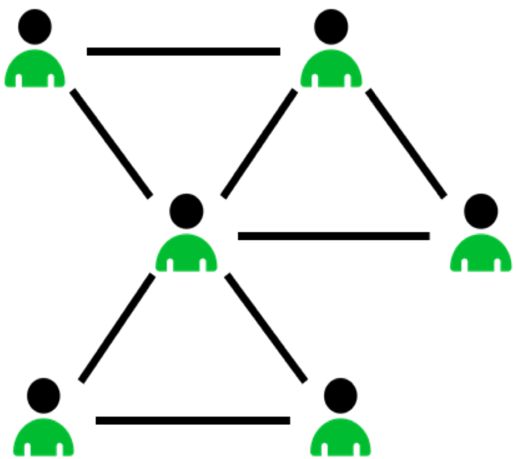
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



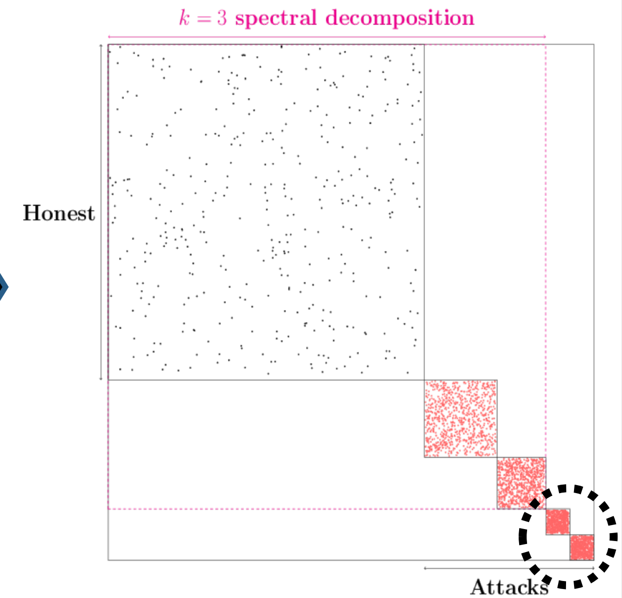
- Represent input graph as adjacency matrix

- Use rank-*k* decomposition to find latent factors associated with fraudulent following behavior

Caveats of decomposition



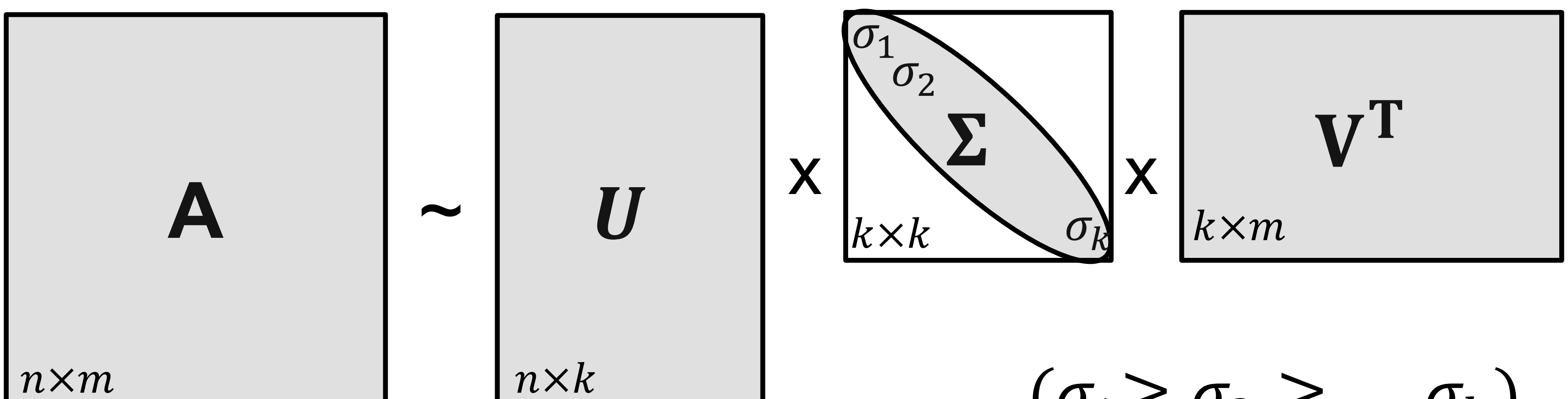
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



- Decomposition methods miss "stealth attacks" below top-*k* factors
- Increasing *k* is computationally expensive

Singular Value Decomposition

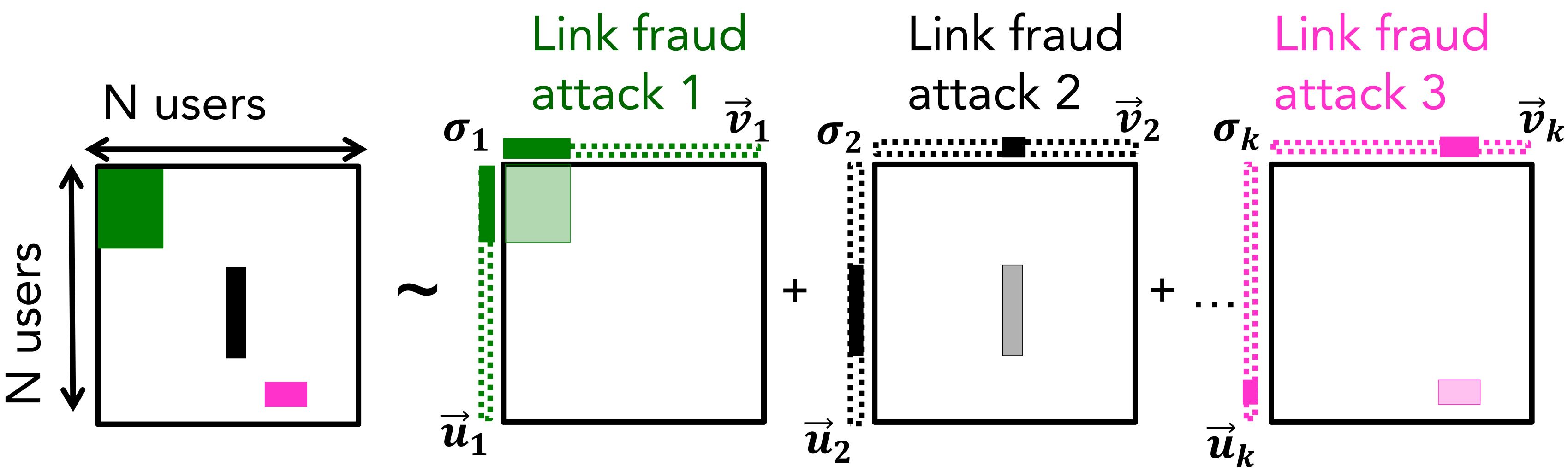
- Used for low-rank matrix approximation
- Rank *k* SVD reduces matrix **A** into *k* latent factors/dense blocks/communities
  - **U** and **V** capture "involvement" of nodes
  - **Σ** denotes factor "strength"



$(\sigma_1 \geq \sigma_2 \geq \ \ldots \sigma_k)$

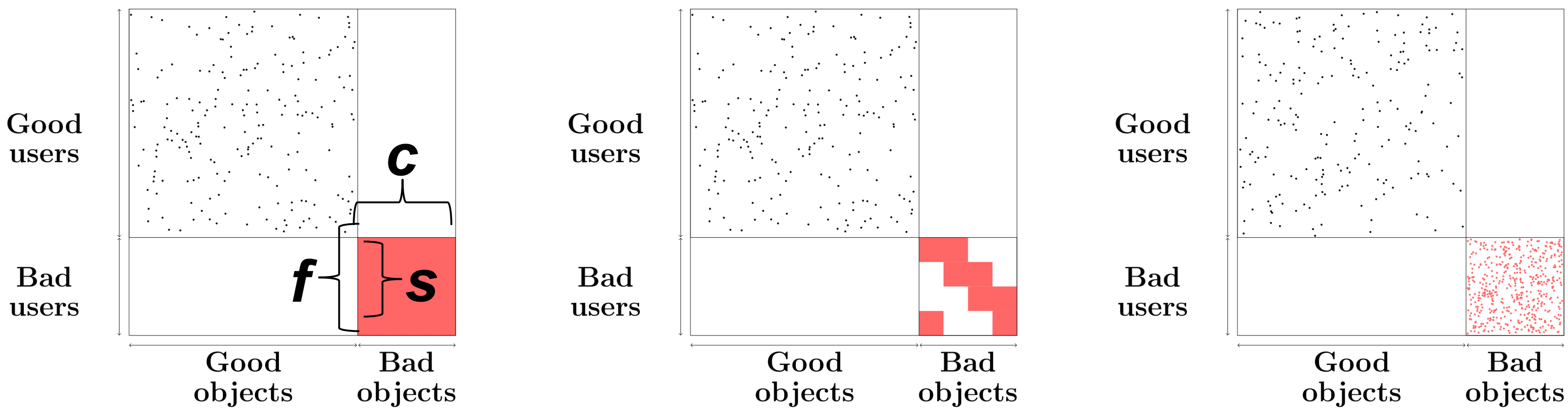
Singular Value Decomposition

- Used for low-rank matrix approximation
- Rank *k* SVD reduces matrix **A** into *k* latent factors/dense blocks/communities
  - **U** and **V** capture "involvement" of nodes
  - **Σ** denotes factor "strength"



SVD: adversarial implications

- Attacker controls **f** fake accounts
- They have **c** customers, wanting **s** links each



$\sigma_1 = \sqrt{cs}$ $\qquad \sigma_1 = s\sqrt{c/f}$ $\qquad \sigma_1 \approx s\sqrt{c/f}$

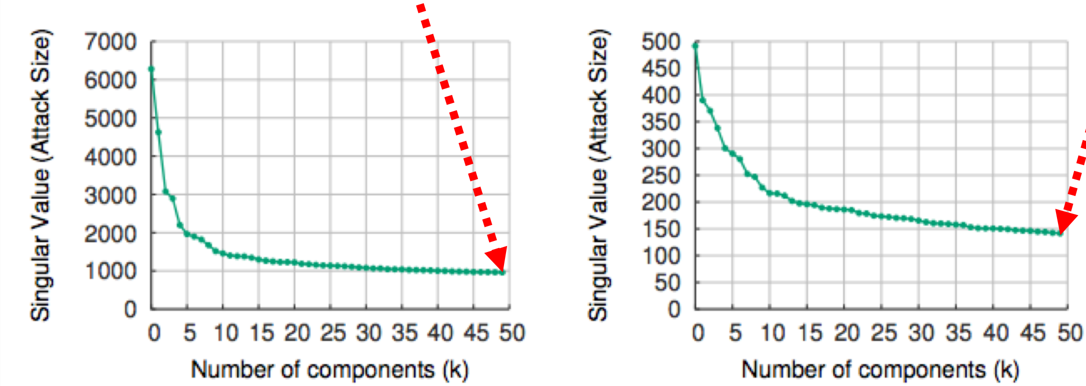
- Attack footprint has a closed form!

Does this even matter? (yes!)

- For $\sigma_k = 50$, attackers could avoid detection while adding…

92K followers to 10 Twitter accounts

140 reviews to 140 Amazon products

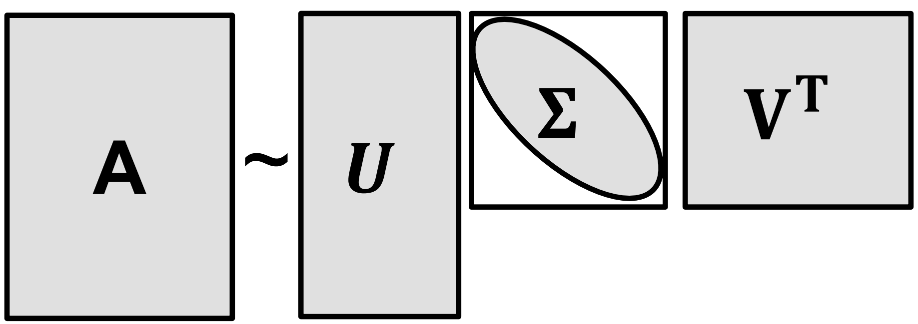


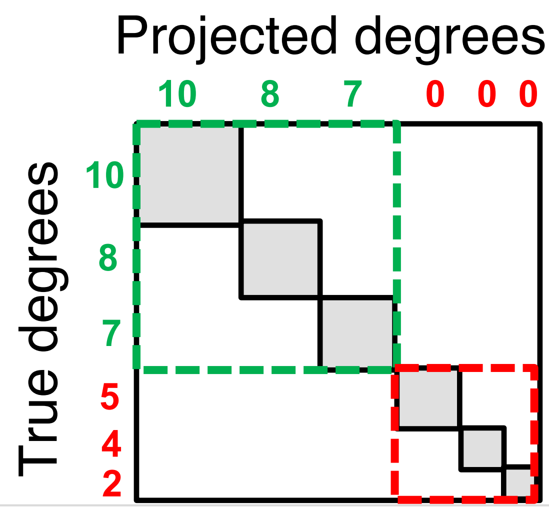
(a) Twitter Followers (b) Amazon Reviews

- So how do we catch them?

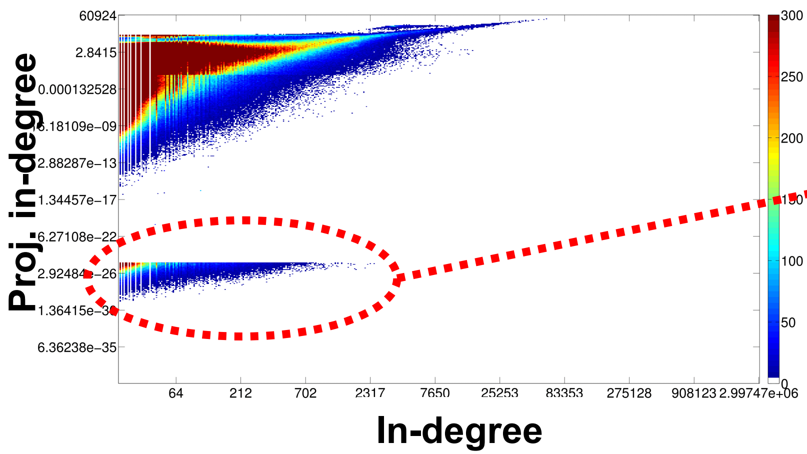
Projection as a signal

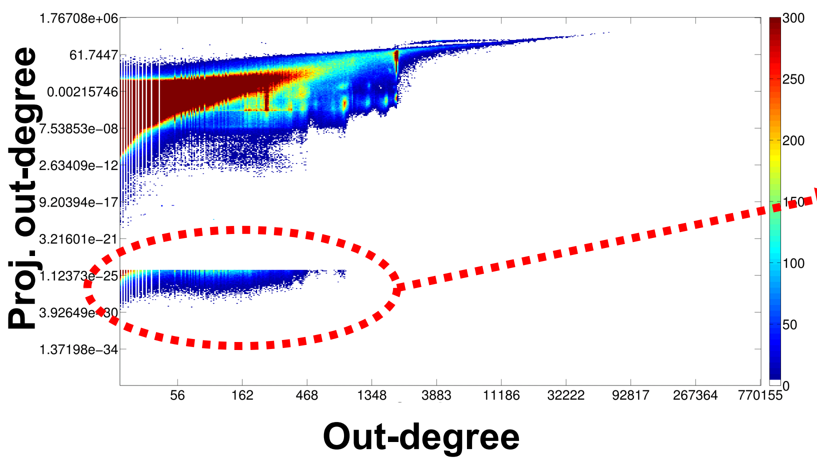
- **Intuition:** Stealth attacks should have very low top-*k* projection, due to poor graph connectivity
- We quantify projection for each node as
  - Projected out-degree: $\|\vec{\boldsymbol{u}_i}\boldsymbol{\Sigma}\|_2^2 \leq \deg_{out}(i)$
  - Projected in-degree: $\|\vec{\boldsymbol{v}_i}\boldsymbol{\Sigma}\|_2^2 \leq \deg_{in}(i)$

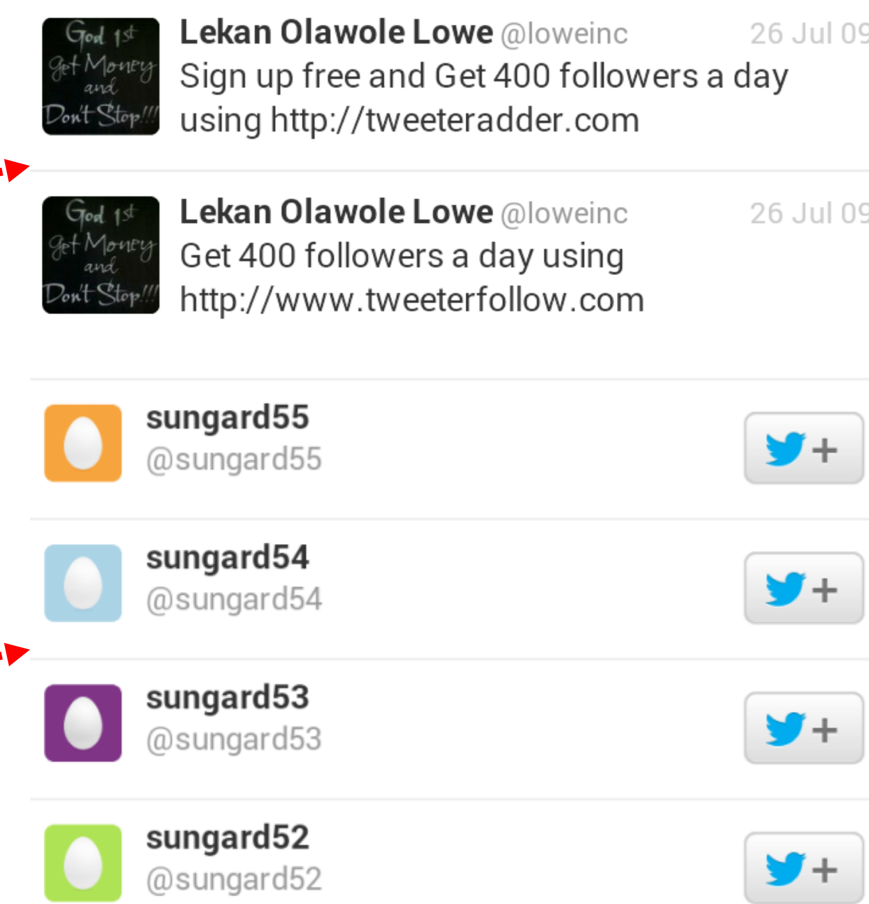




Too-low projection is suspicious







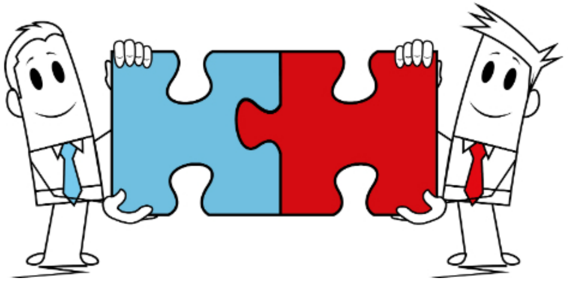
41.7M users, 1.5B edges

Our approach: *FBOX*

- Three basic components
  - Compute rank-k SVD of $\mathbf{A}$
  - Compute true and projected in/out degrees
  - Identify nodes with too-low projection with respect to peers as suspicious

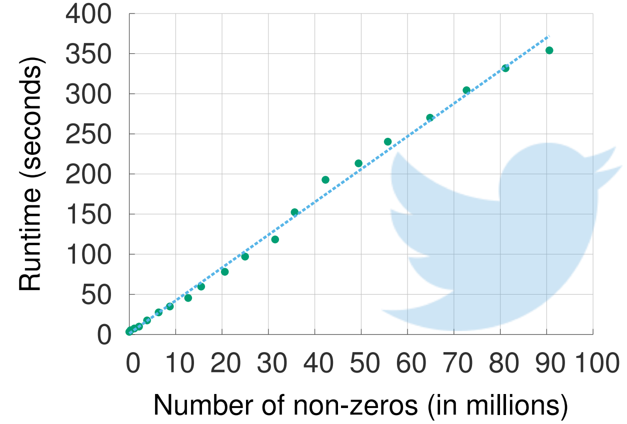
- *FBOX* complements existing spectral methods

Code publicly available at:
https://goo.gl/gcQMvS



Experimental results

- ▣ *93%* precision in manual validation experiment
- ▣ *70%* of suspects were previously uncaught by Twitter, and had engaged in misbehavior for years
- ▣ *83% precision* on synthetic attacks with half camouflage links
- ▣ Linear scaling on # edges



Technical insights

- Simple relationship structure can be well-exploited to identify fake engagement behaviors

- Dimensionality reduction can help “prime” structured data for outlier detection

- Summary statistics depend on sample size → affects data distribution and outlier detection

Two examples

- Spotting suspicious link behavior in online social networks

- **Combating fake viewership on livestreaming platforms**

What is livestreaming?

- Livestreaming connect *viewers* with *channels*
- Streamers own channels, go live at their whim and *broadcast* content

YouTube Live

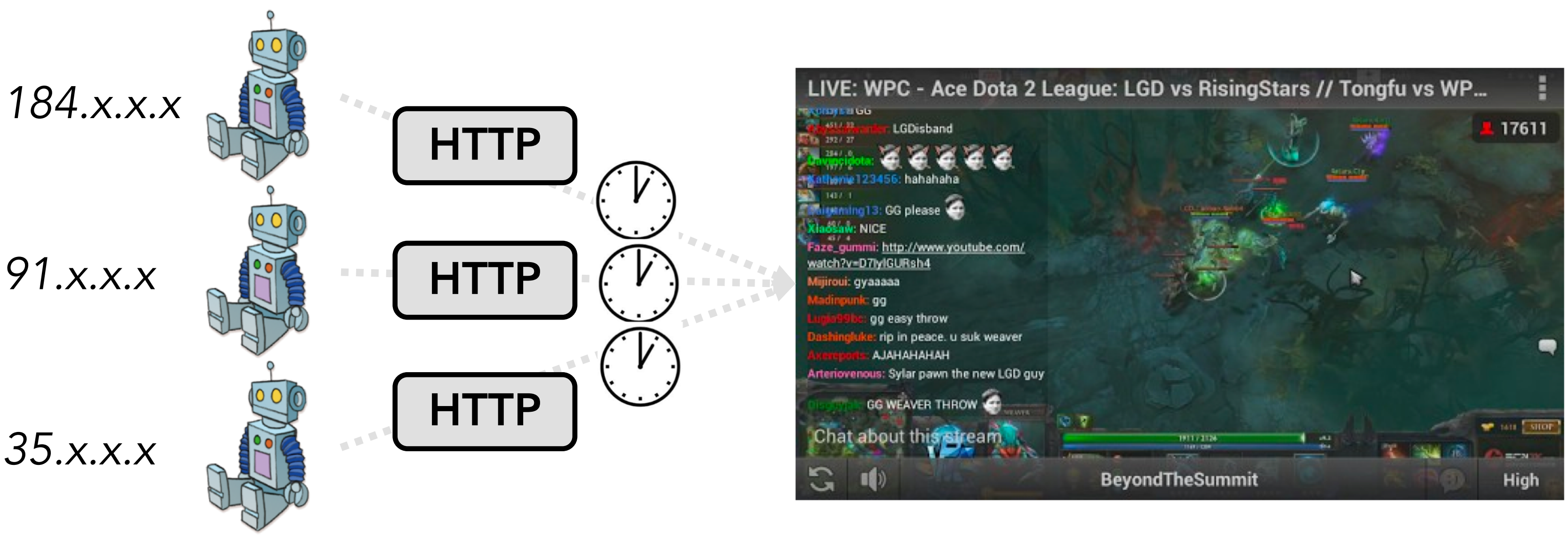
livestream

USTREAM

twitch

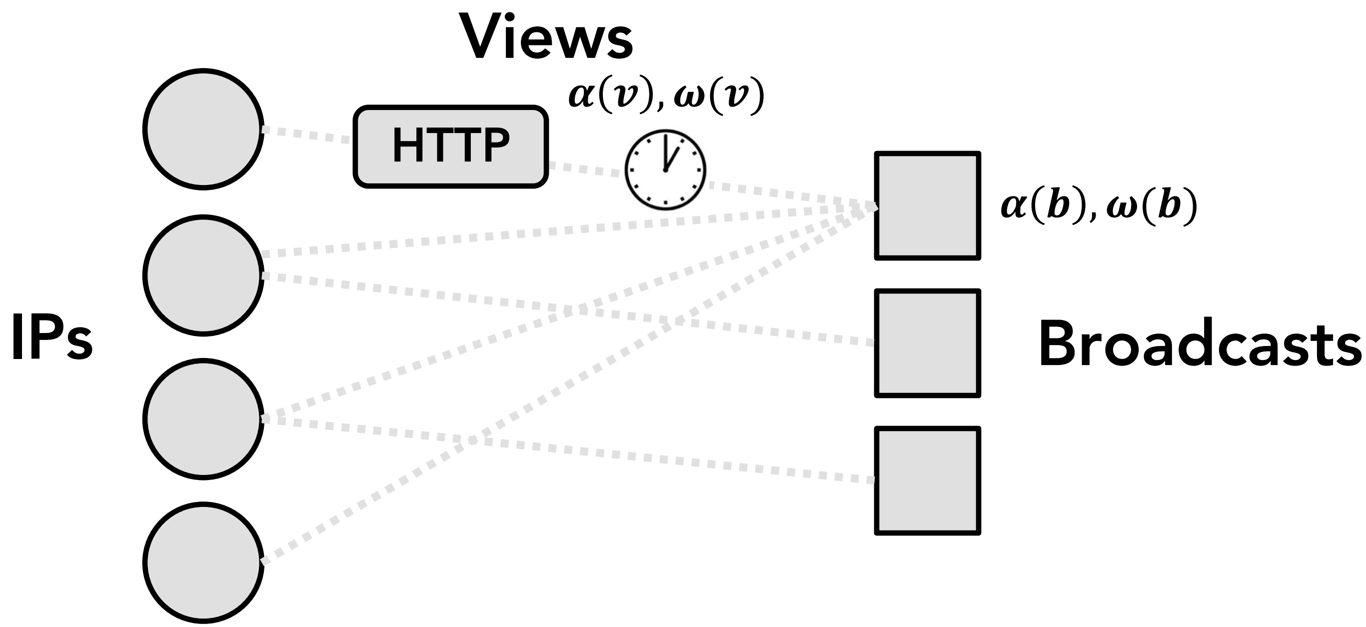
Viewbotting on livestreaming

- Live viewership is the key popularity metric
- Faking viewer count offers monetization and recommendation benefits
- Accomplished via "phantom" viewbots



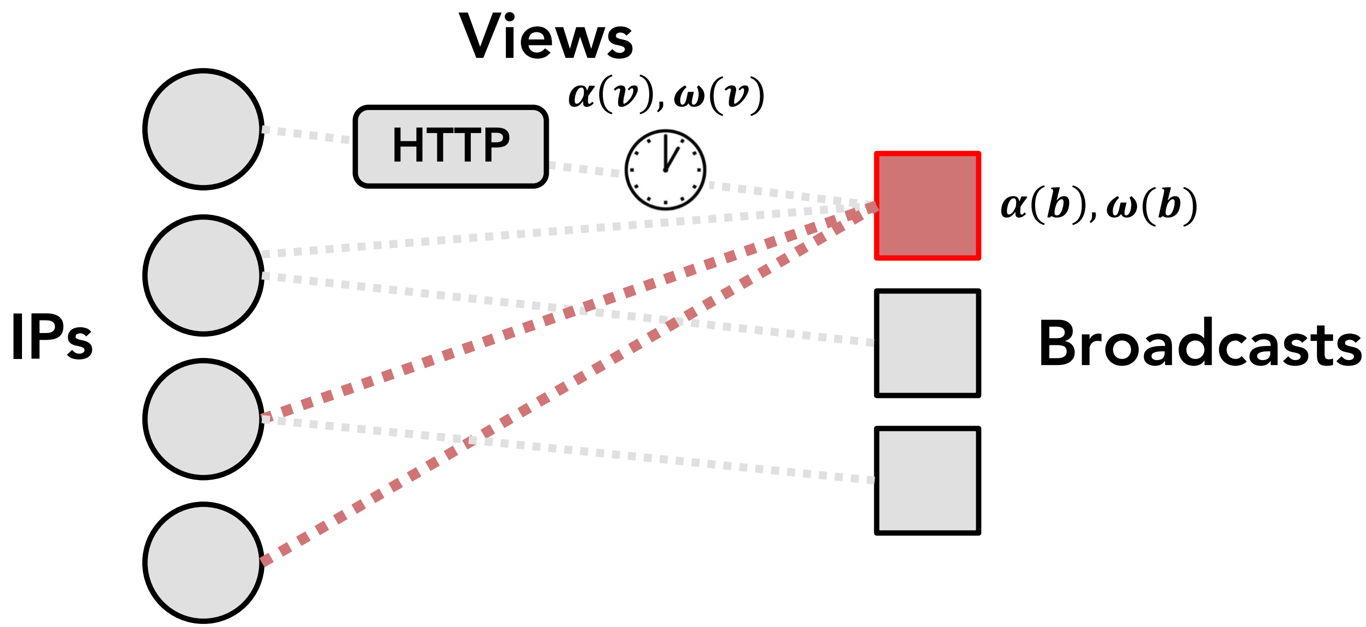
Problem definition

Given: *views V to broadcasts B (many-to-one)*



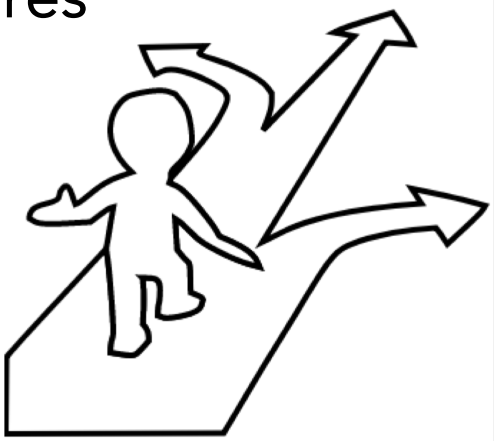
Problem definition

Given: *views* V *to broadcasts* B *(many-to-one)*
Find: *viewbotted broadcasts* B_{botted} *and constituent botted views* V_{botted}



Approach considerations

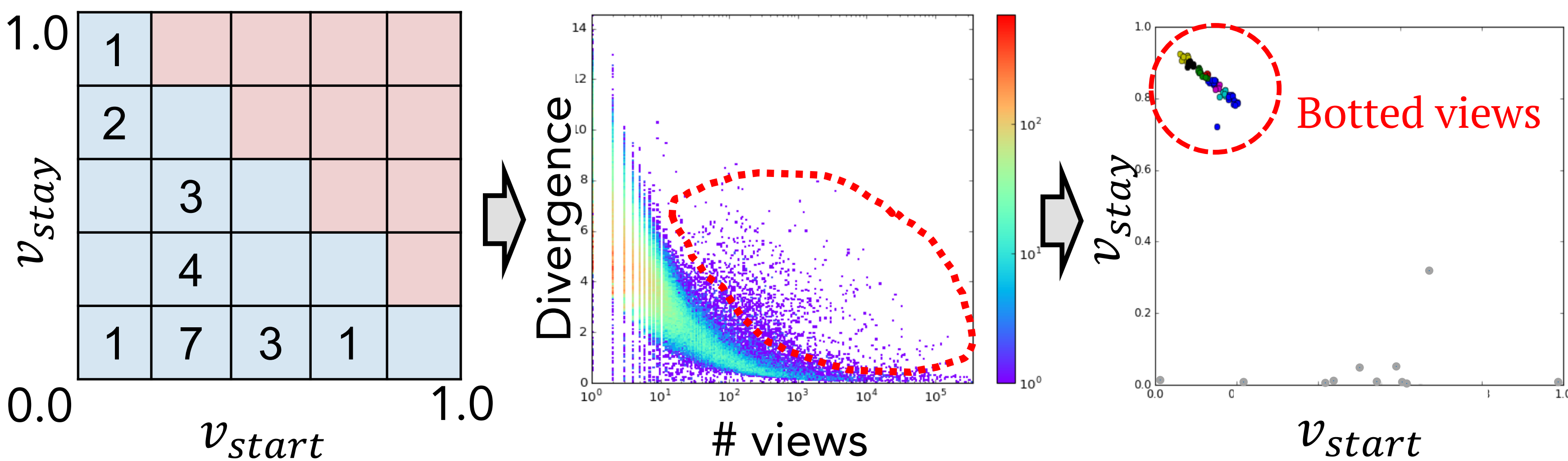
- Problem constraints
  - No labels/ground truth
  - Only have HTTP and timestamp features



- Resulting choices
  - Unsupervised approach
  - Focus on groups of views instead of individuals
  - Target temporal features – harder to spoof and directly related to attacker constraints

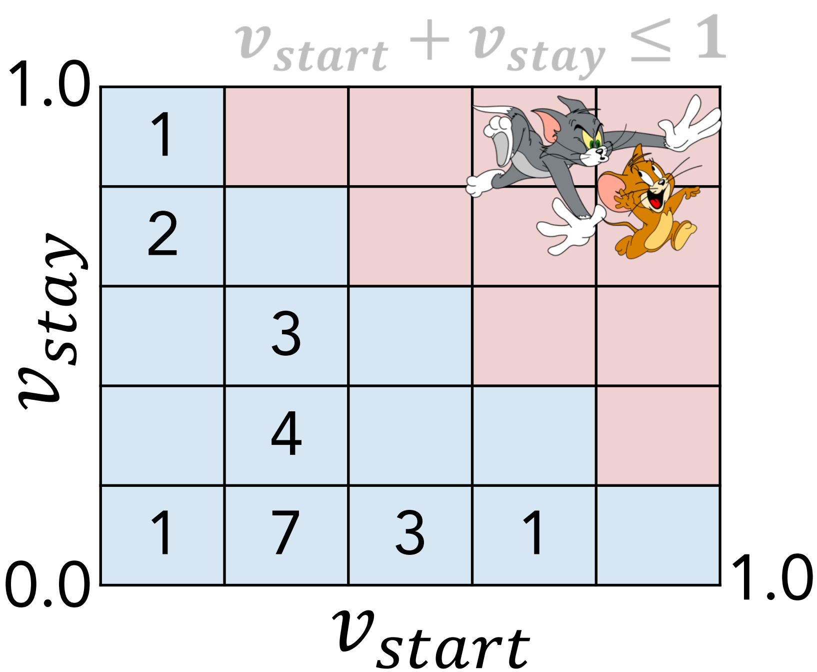
Our approach: *FLOCK*

- Three basic components
  - Modeling broadcast viewership
  - Identifying viewbotted broadcasts
  - Identifying fake views



Modeling broadcast viewership

- Broadcasts are not mathematical objects
- But we can model them as such: “bag-of-views”

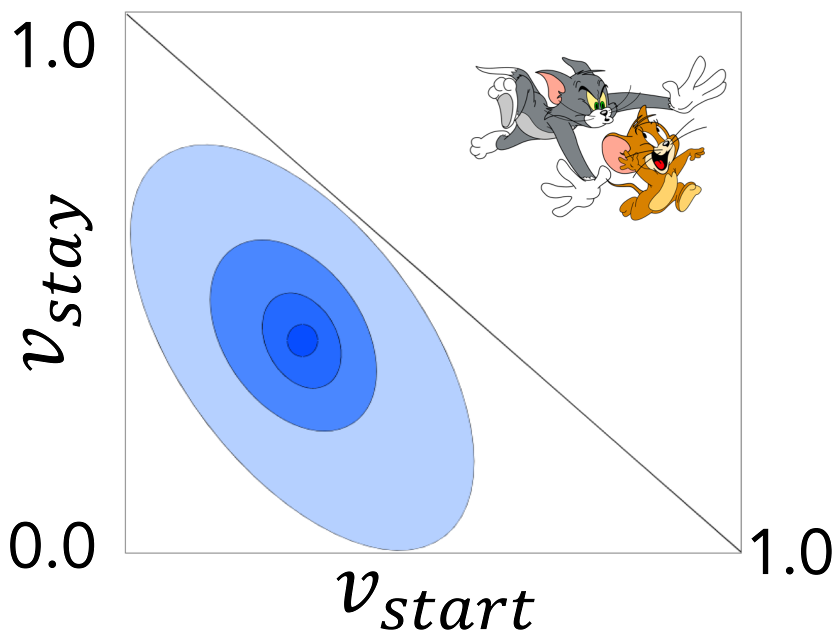


$$v_{start} = \frac{\overset{\text{View start}}{\alpha(v)} - \overset{\text{Broadcast start}}{\alpha(b)}}{\underset{\text{Broadcast end}}{\omega(b)} - \underset{\text{Broadcast start}}{\alpha(b)}}$$

$$v_{stay} = \frac{\overset{\text{View end}}{\omega(v)} - \overset{\text{View start}}{\alpha(v)}}{\underset{\text{Broadcast end}}{\omega(b)} - \underset{\text{Broadcast start}}{\alpha(b)}}$$

Modeling broadcast viewership

- Broadcasts are not mathematical objects
- But we can model them as such: "bag-of-views"



View start Broadcast start

$$v_{start} = \frac{\alpha(v) - \alpha(b)}{\omega(b) - \alpha(b)}$$

Broadcast end Broadcast start

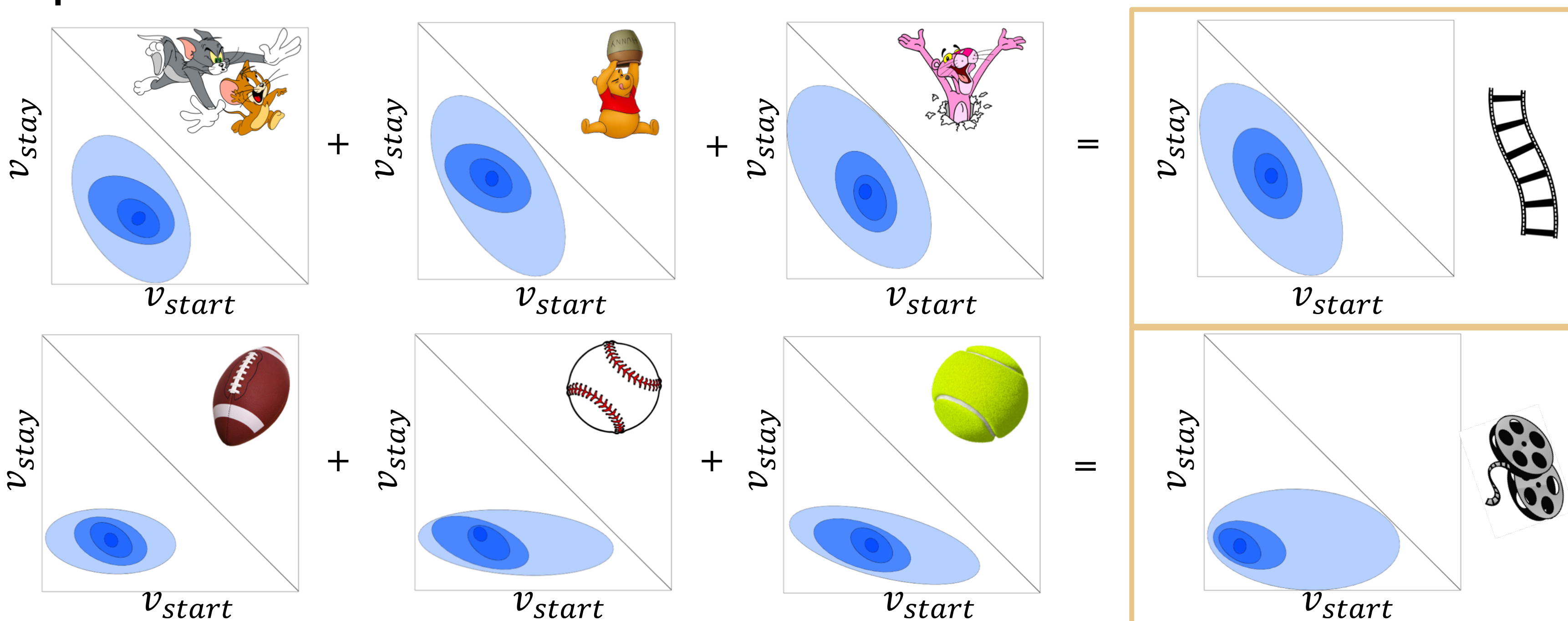
View end View start

$$v_{stay} = \frac{\omega(v) - \alpha(v)}{\omega(b) - \alpha(b)}$$

Broadcast end Broadcast start

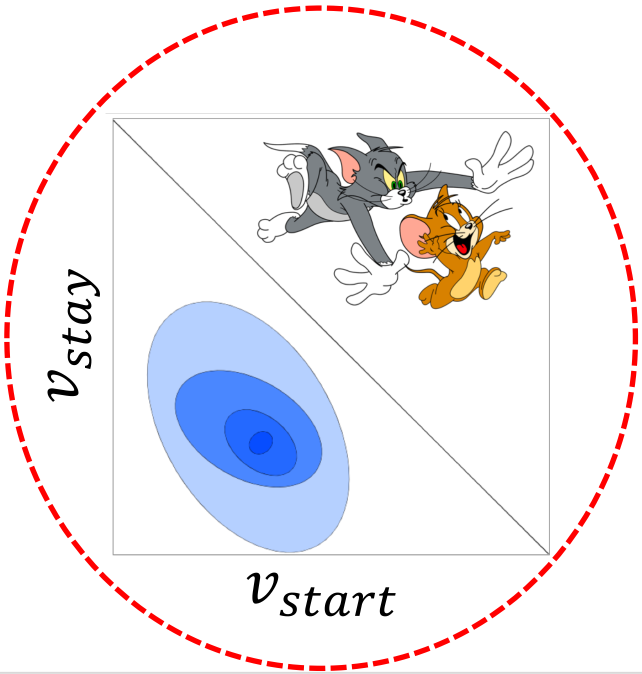
Modeling broadcast viewership

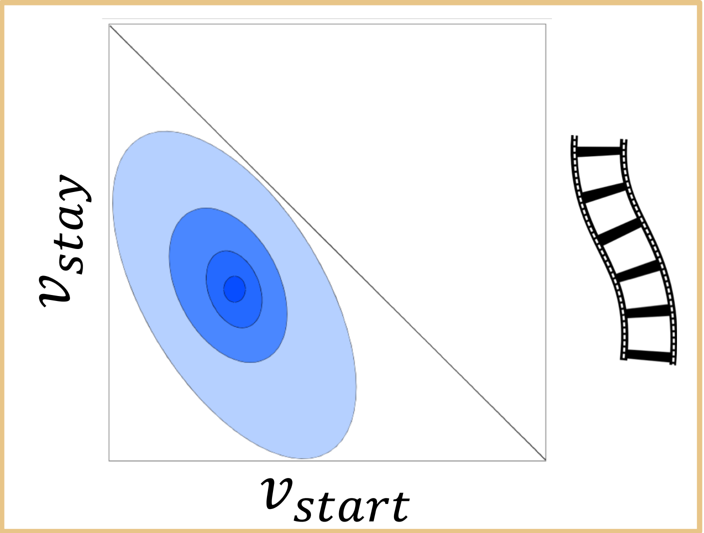
- We can model "typical" viewership across many broadcasts via multinomial MLE, but…
- Duration influences behavior → create duration-specific *bracket* distributions



Modeling broadcast viewership

- **Intuition:** bracket distributions describe "typical" broadcast viewership behavior
- They enable us to evaluate "closeness" of a broadcast with respect to the bracket

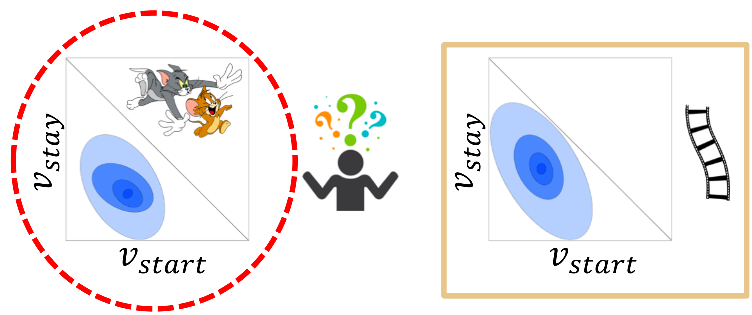




Identifying viewbotted broadcasts

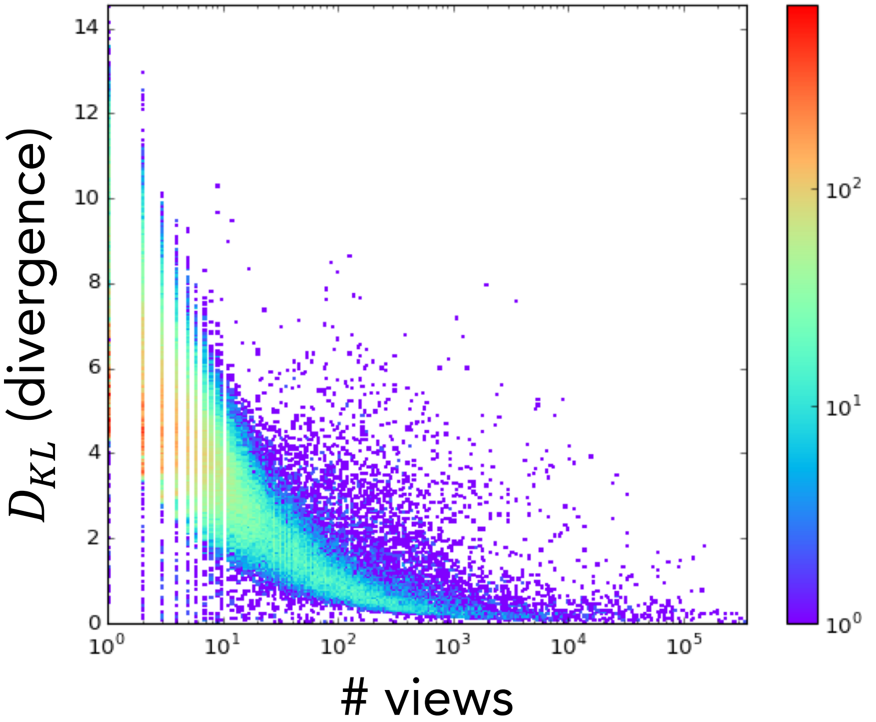
- We can measure closeness using distributional distance measures
- We use Kullback-Leibler (KL) divergence between broadcast b and bracket $\beta(b)$

$$D_{KL}\big(\underbrace{b}_{\text{Broadcast}} \parallel \underbrace{\beta(b)}_{\text{Bracket}}\big) = \sum_{\underbrace{i}_{\text{Outcome}}} b_i \cdot \log_2 \frac{b_i}{\beta(b)_i}$$



Identifying viewbotted broadcasts

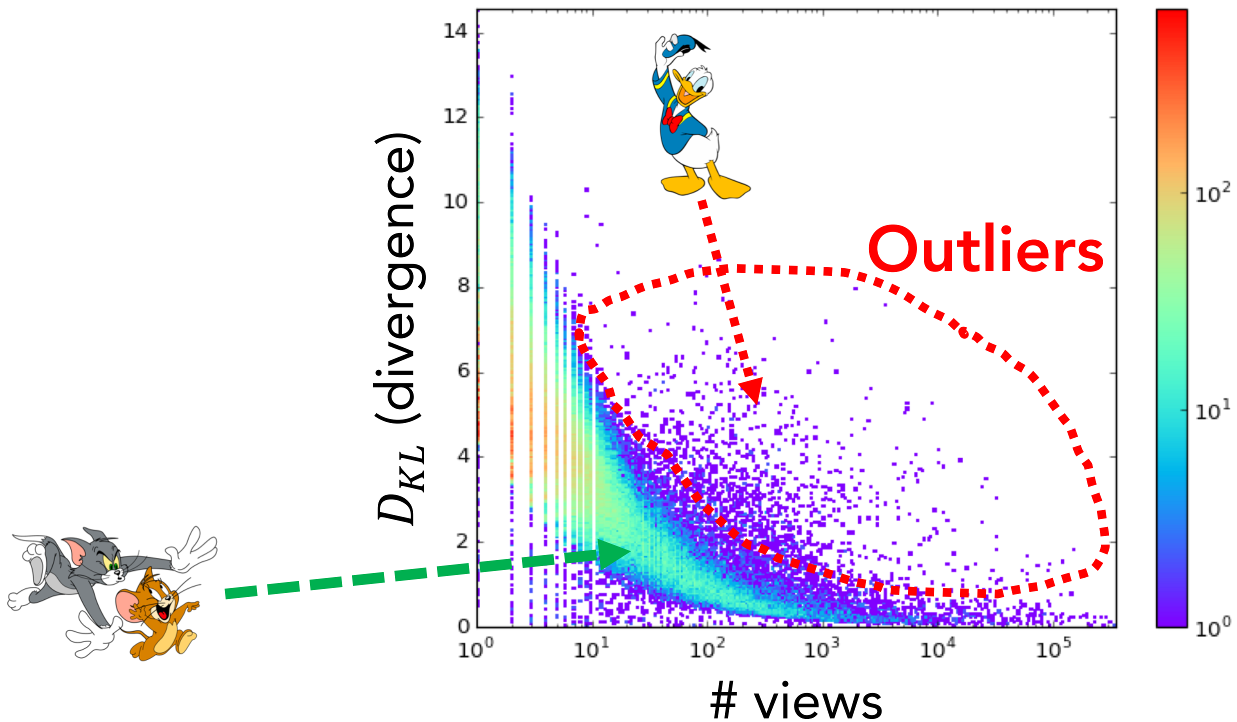
- Most broadcasts are close to brackets



16M views, 100K broadcasts

Identifying viewbotted broadcasts

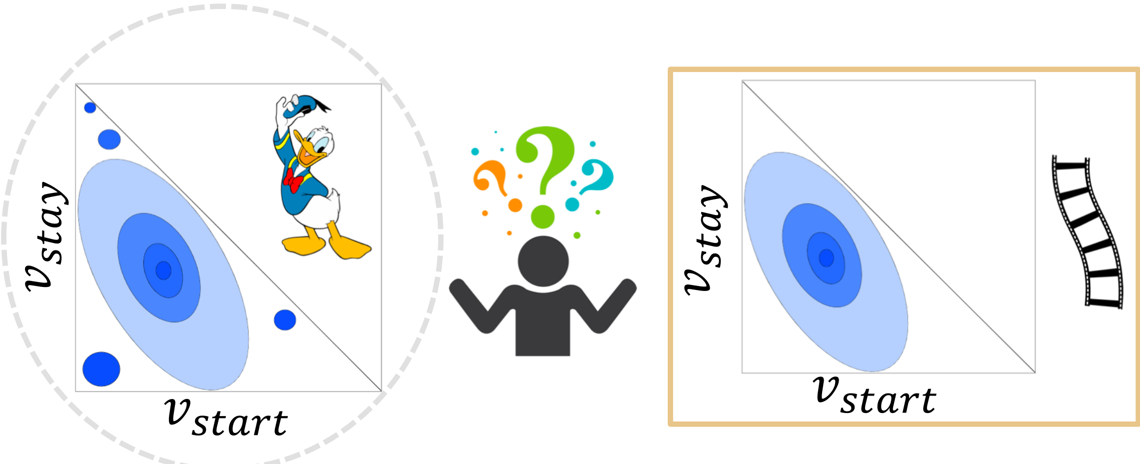
- Most broadcasts are close to brackets



- Too-high divergence w.r.t. #views → suspicious

Identifying fake views

- Broadcasts are outlying because they have suspicious views → which ones?

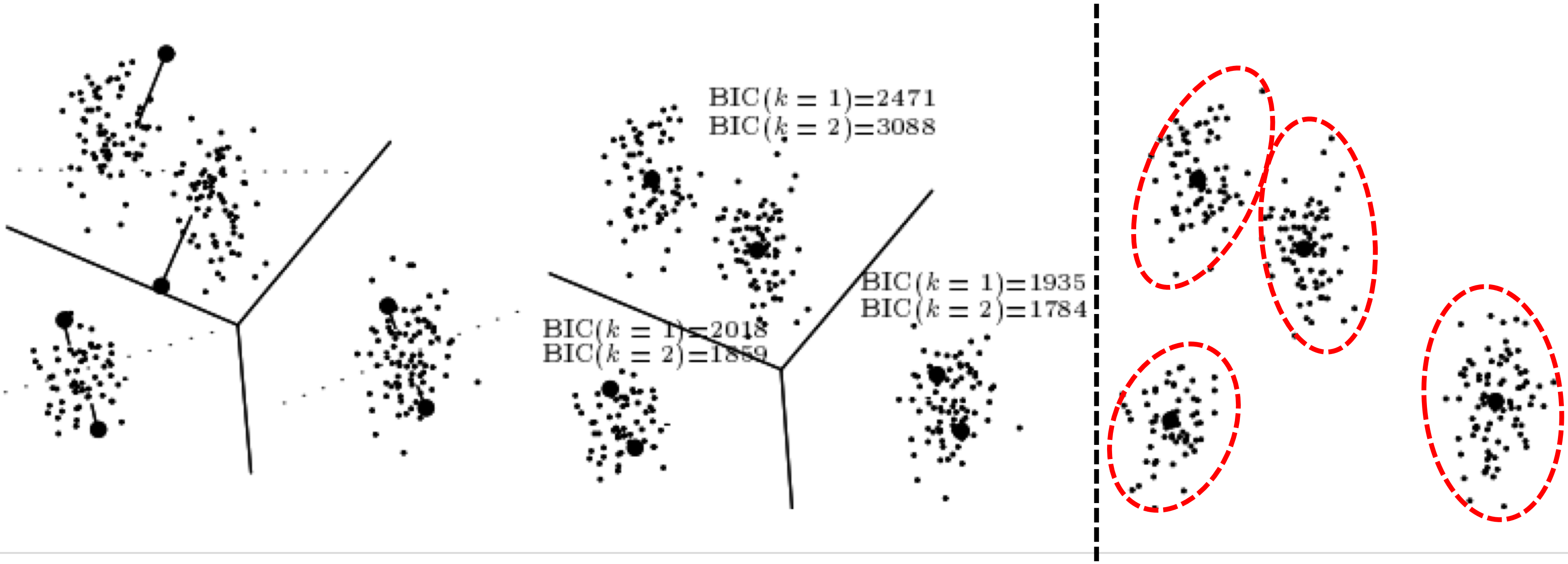


- **Intuition:** Find clusters causing high divergence
  - How do we *cluster* the views?
  - How do we *choose* the right clusters?

Identifying fake views: *clustering*

- Could use any general $\mathbb{R}^n$ clustering solution

- Since we don't know # clusters a priori, we use non-parametric clustering (*Pelleg et al, 2000*)



Identifying fake views: *choosing*

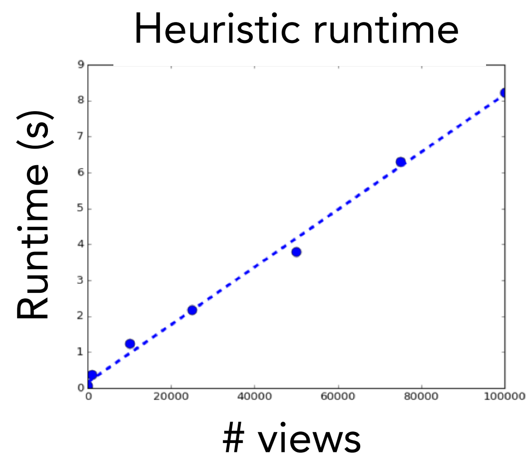
- D_{KL} should shrink when bad clusters are removed, since viewership is more "typical"

$$\min_{b' \in 2^C} D_{KL}\left(b' \parallel \beta(b')\right)$$

Powerset of clusters

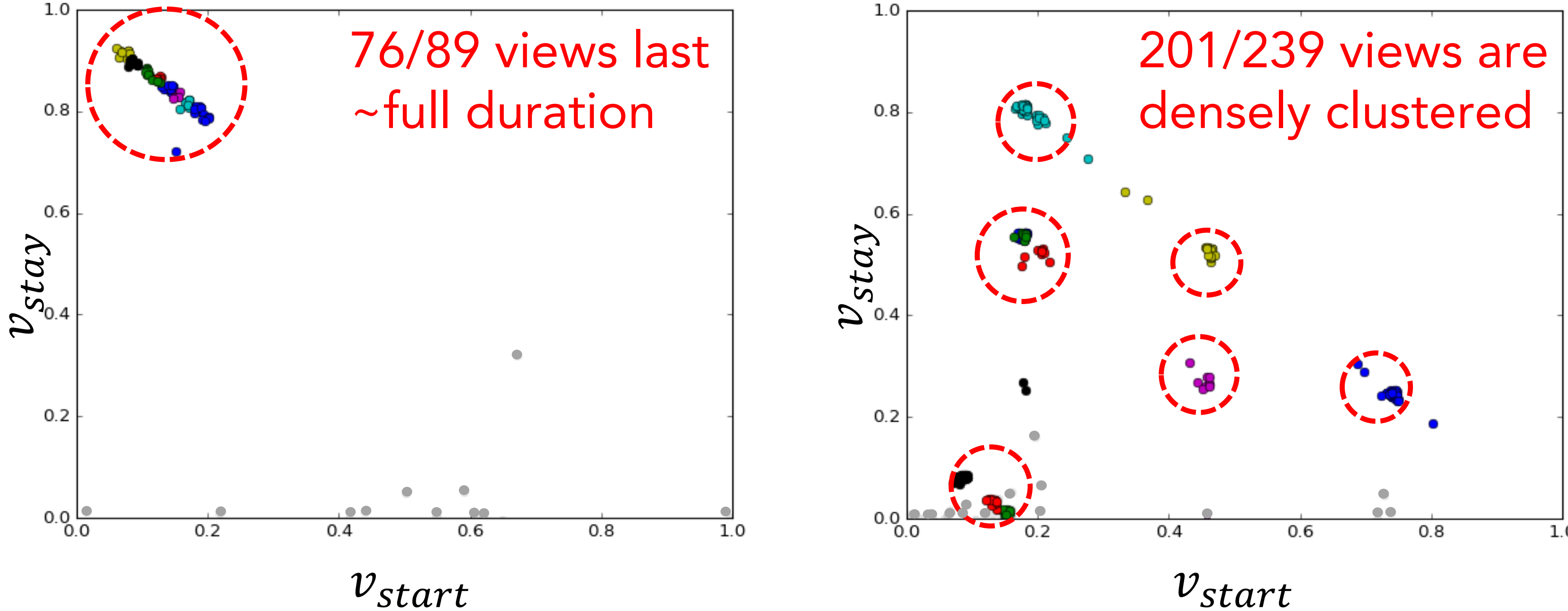
Pruned broadcast

Bracket



- Since this objective is intractable for large C, we propose a greedy approach
  - Deterministic, guaranteed to converge

Experimental results: *broadcasts*

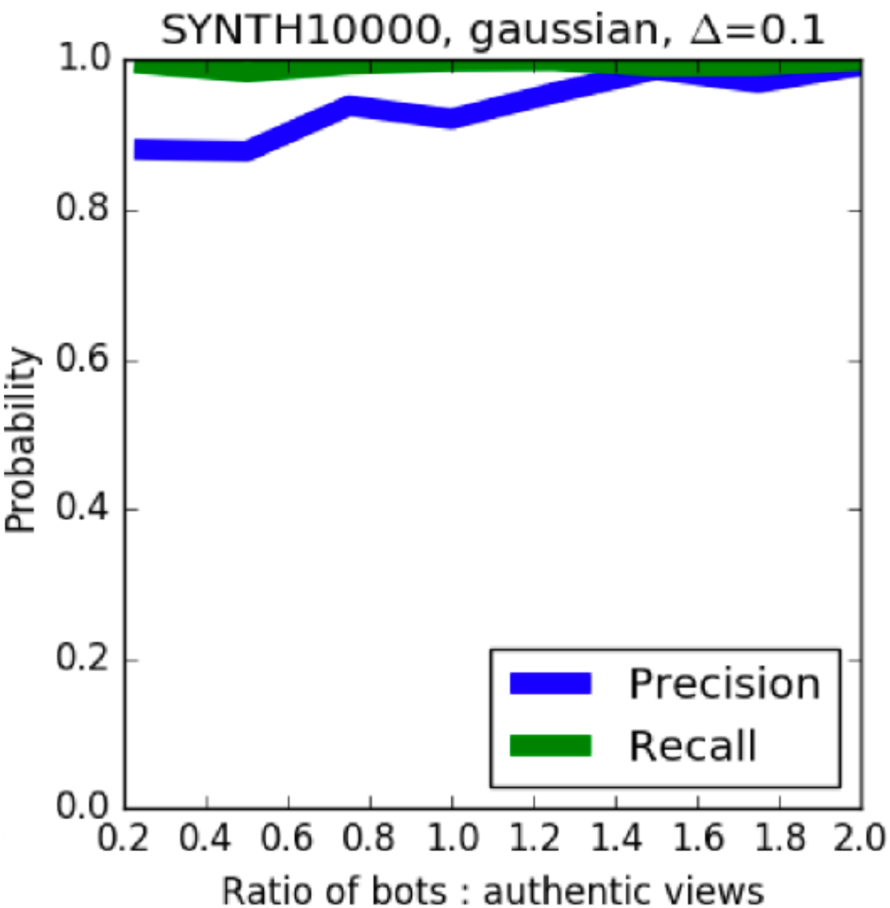


- 98% positive and 99% negative precision in manual broadcast labeling task
- Broadcasts labeled according to ISP/IP regularity in views

Experimental results: *views*

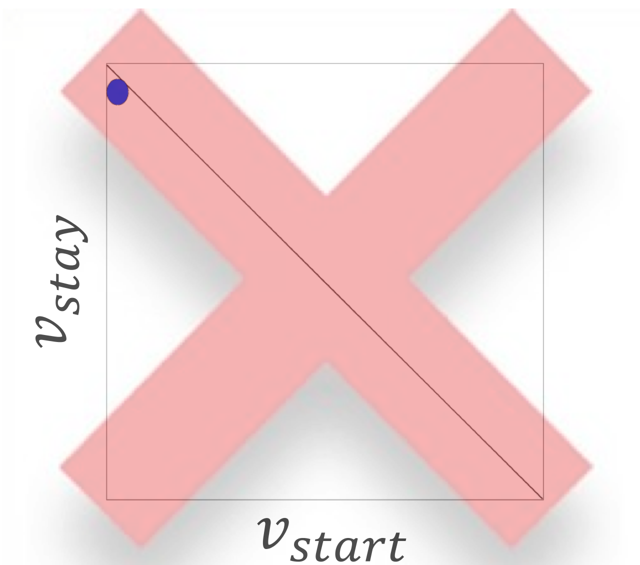
- Synthetic attacks with varying parameters
  - Ratio of "good" and "bad" views
  - Temporal "bad" view distribution

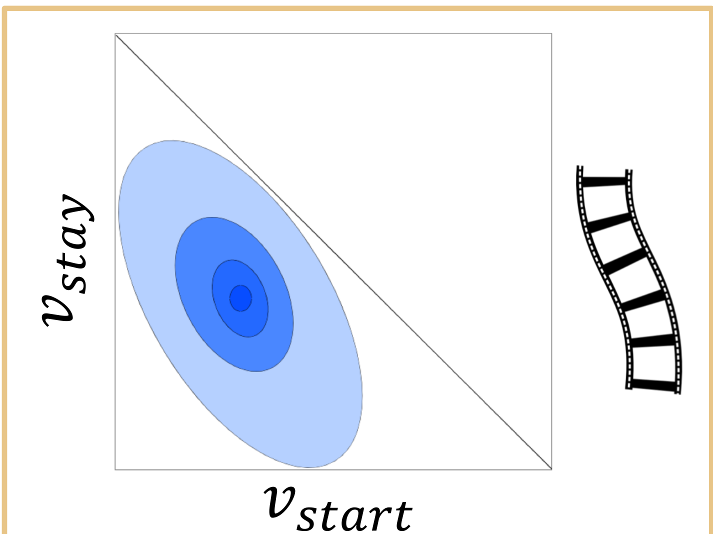
- *95%* precision and *100%* recall



Adversarial implications

- Even if an adversary knows the right bracket and target distribution, they still need *40%* more IPs than under naïve rate-limiting to do as well





Technical insights

- Real data can be structurally complex; distributions can be more suitable than points

- Some outlying phenomena are only meaningfully outlying in groups

- Hierarchical outlier detection can reduce problem complexity

Back to the bigger picture

- **We can tackle a wide variety of misbehavior detection tasks by identifying the right types of outlying users.**

- Outlier detection plays an important role in the detection of misbehavior

    …and many other application areas!

Tempering expectations

- **We can tackle a wide variety of misbehavior detection tasks by identifying the right types of outlying users.**

- But outlier detection is *not* a "silver bullet"
  - Is outlier detection the best solution for this task?
  - How should my task influence my detection strategy?
  - Are the detected outliers relevant to my task?

Remark: Suitability

Not all problems are best-suited for outlier detection.

"If all you have is a hammer, everything looks like a nail." – Maslow's hammer

Crowdsourcing

Classification & automated response

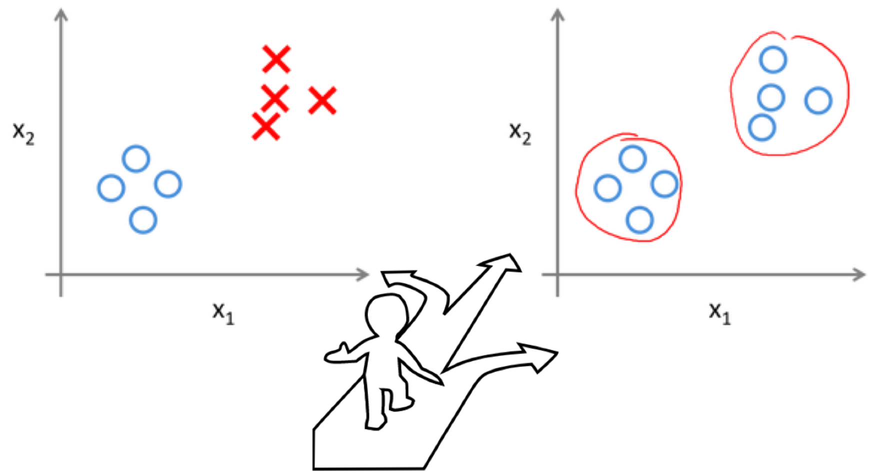
Revisiting incentive structure

Changing platform infrastructure

Remark: Problem-specificity

Outlier detection strategies can be highly problem-specific.

Parametric or non-parametric	*Group-wise or individual*
Multivariate or univariate	*Hierarchical or flat*
Distributions or point values	*Online or offline*



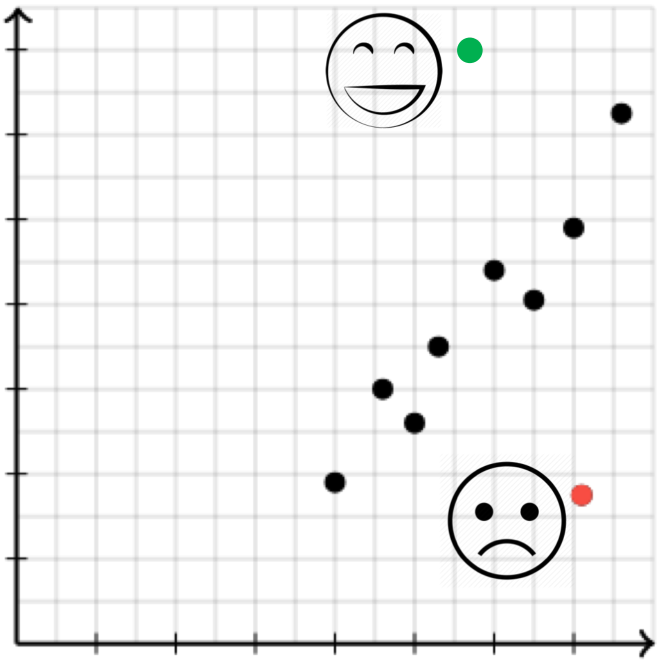
Remark: Value

An outlier is only as valuable as the behavior it indicates.

Fake follower or incompetent Twitter user?

Malicious user or hacked account?

Fake news article or satire?



Implications

Outlier detection in practice should…

- ▣be well-justified in motivation
- ▣be tailored to address problem constraints
- ▣be vetted to actually solve *that* problem with minimal error

Snap is hiring!

- Research Scientists/Engineers/Interns in Security, Data Mining, Deep Learning, NLP, HCI, Graphics, Vision & Computational Imaging

- Many opportunities to work w/ academics

- Reach out if you're interested in collaborating!

nshah@snap.com

http://www.cs.cmu.edu/~neilshah

Snap Inc.