Carnegie Mellon University

中国科学院 CHINESE ACADEMY OF SCIENCES

HOLOSCOPE: TOPOLOGY-AND-SPIKE AWARE FRAUD DETECTION

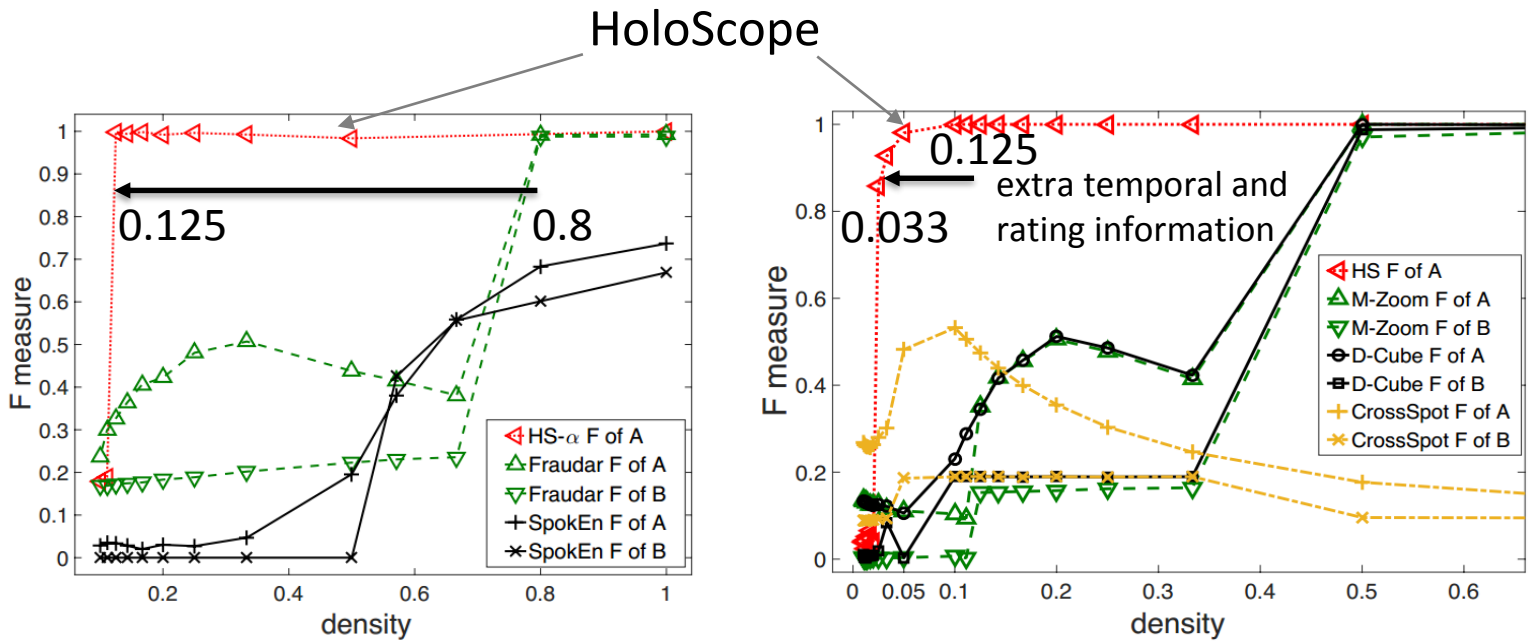
Shenghua Liu+

Joint work with Bryan Hooi*, Christos Faloutsos*

Email: liu.shengh@gmail.com

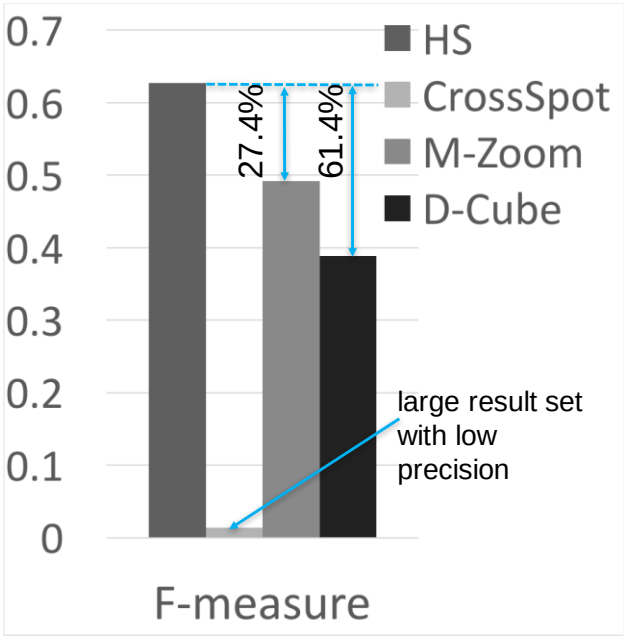
HoloScope: Topology-and-Spike Aware Fraud Detection

- Our HoloScope: HS-α and HS detect injected fraudsters with **higher accuracy** (F measure), even when the injection density become lower.



HoloScope: Topology-and-Spike Aware Fraud Detection

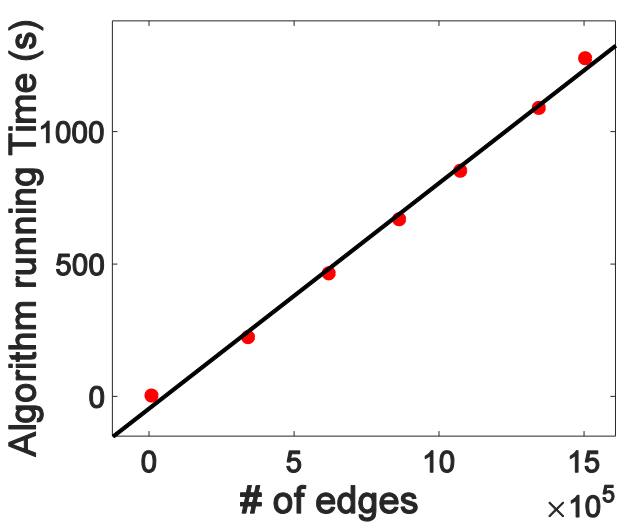
- Our HoloScope: HS detects suspicious users in online system data (Microblog: Sina Weibo).



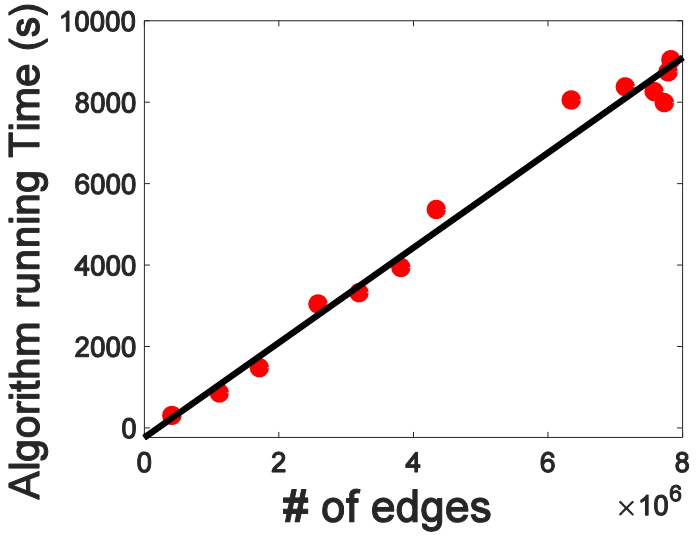
2.75 M users, 8.08 M messages, and 50.1 M edges in our data of Dec 2013

HoloScope: Topology-and-Spike Aware Fraud Detection

- Our HoloScope: runs near-linear time in # of edges.



BeerAdvocate dataset



Amazon Electronics dataset

Outline

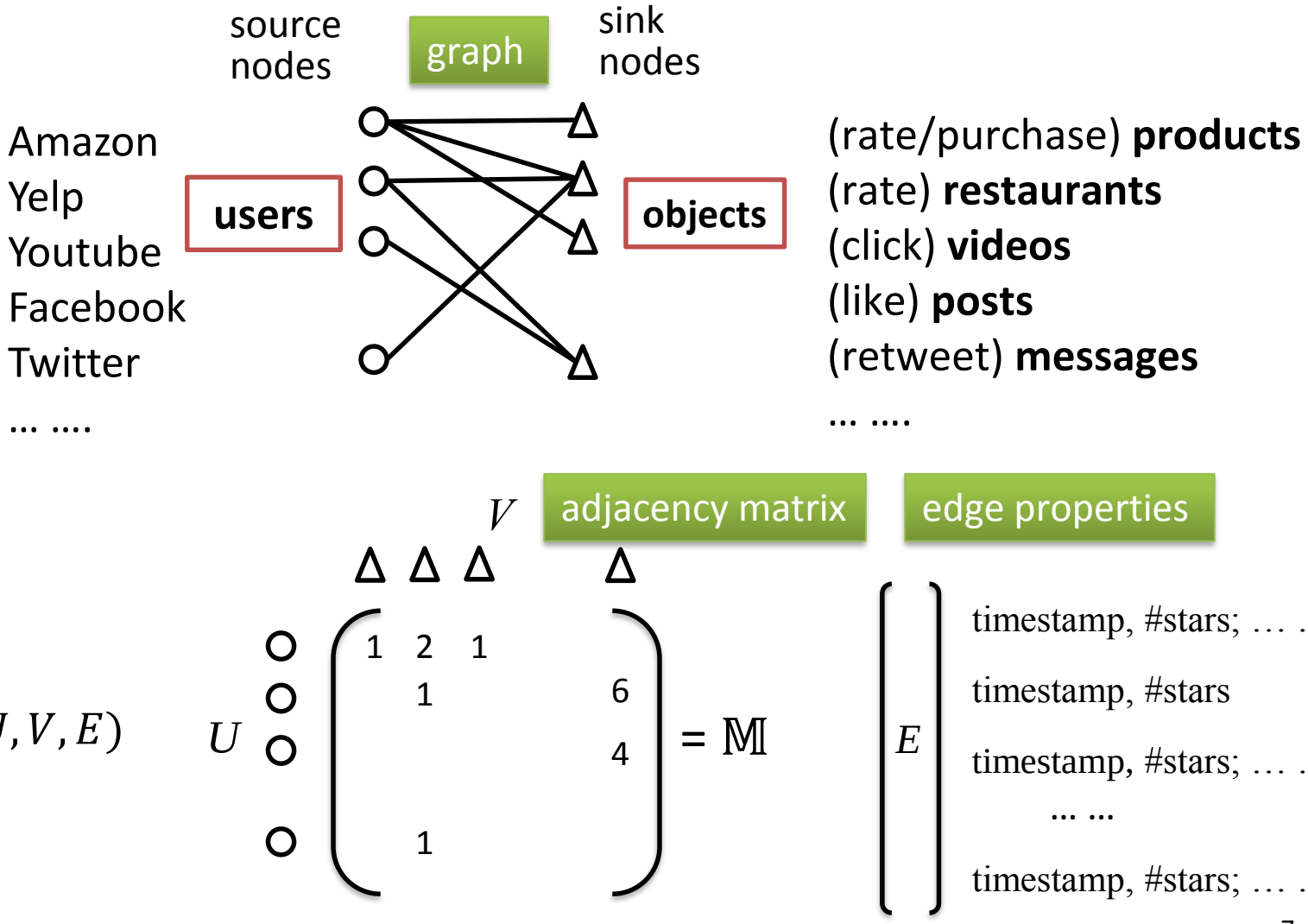
- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
- Experiments
- Conclusion

Online activities affects the business

- Users make decisions about
  - which to buy, where to eat and live
- Recommendation of
  - news (Facebook newsfeed), videos (Youtube)
- Merchants pay for advertisement and commission

- All rely on: # of sales, # of clicks and likes, and # of stars

Abstract activities into bipartite Graph



Problem of fraud detection

- **Given:**
  - (user, object, timstamp, #stars)
- **Find:**
  - a group of suspicious users, and objects,
- **To optimize:**
  - the metric of suspiciousness from topology, rating time and scores.

(user, object, timestamp~~, #stars~~)

(user, object, ~~timestamp,~~ #stars)

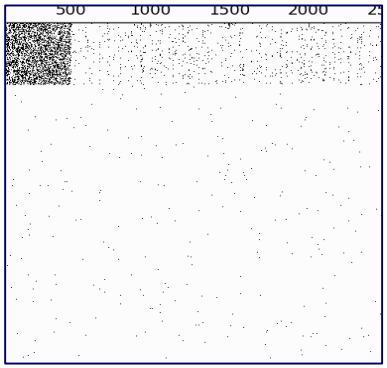
(user, object~~, timestamp, #stars~~)

Outline

- Background and Problem
- Graph-based fraud detections
- HoloScope Algorithm
- Experiments
- Conclusion

Why using graph to detect fraud?

- Content can be cheated by NLP technology
- Content is not available
- Graph is a good representation of
  - users reviewing/giving scores to objects
  - a user clicking a link, and watching a video
- Dense blocks in such a graph are usually suspicious



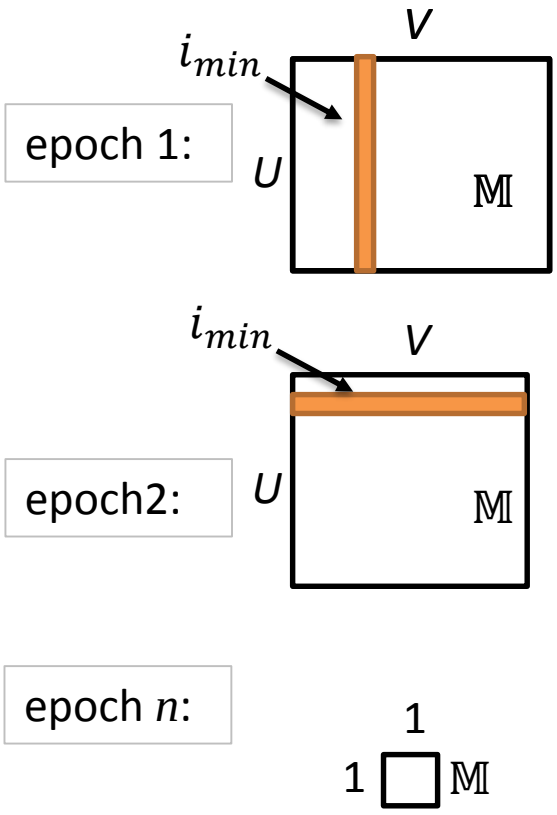
Average degree density works better than volume density for fraud detection

- Volume density
  - Suppose
    - ✓a fraudster has # of accounts: a
    - ✓his goal is click b objects 200 times
  - Density: $(b \cdot 200)/(a \cdot b) = 200/a$
  - unlimited b does not increase density
- Average degree: arithmetic / geometric
  - Arithmetic avg: $(b \cdot 200)/(a + b)$
  - Geometric avg: $(b \cdot 200)/\left(\sqrt{ab}\right)$

[Asahiro et al, SWAT'96] [M Charikar, 2000] [B Hooi et al, KDD'16]

A near-linear heuristic algorithm to detect dense block

- **Given**: adjacency matrix $\mathbb{M}$
- $X \leftarrow \{U, V\}$
- **While** X is not empty
  - $i_{min} \leftarrow \arg\min_{i \in X} \deg(i, X)$
  - $X \leftarrow X \setminus \{i_{min}\}$
  - Keep X_{best} that has the best arithmetic avg degree $g(X_{best})$
- **Return** X_{best}

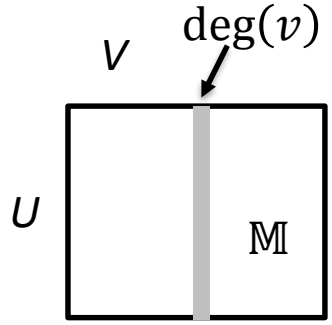


- Theoretical boundary: $g(X_{best}) > \frac{1}{2} g(X_{opt})$
- Time complexity with Priority Tree: $O(|E| \log(|V| + |U|))$
- Optimal algorithm needs $O(|V|^2 \log^2 |V|)$

[A.V. Goldberg, Technical report, 1984]
[Asahiro et al, SWAT'96] [M Charikar, 2000] [B Hooi et al, KDD'16]

Very popular products are less suspicious

- Fraudar penalizes the weight of each edge in $\mathbb{M}$
  - preprocess: $e_{uv} \leftarrow 1/\log(\deg(v) + c) \cdot e_{uv}$,
    - ✓ where $e_{uv} = \mathbb{M}(u,v)$, $c=5$
  - avg degree: $g_{log}(X) = \frac{1}{|X|}\sum_{u,v \in X} e_{uv}$



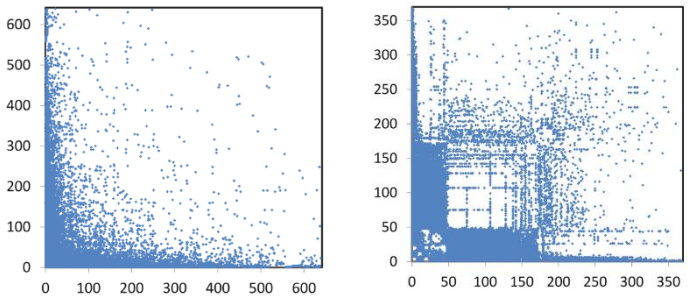
- Bounding Fraud
  - Upper bound of fake edges that m_0 fraudsters can create for n_0 products:

$$2(m_0 + n_0) \cdot g_{log}(X_{best}) \cdot \log(\frac{m_0}{\lambda} + c)$$

    - ✓ where λ is the fraction of edges that an object has from fraudsters.

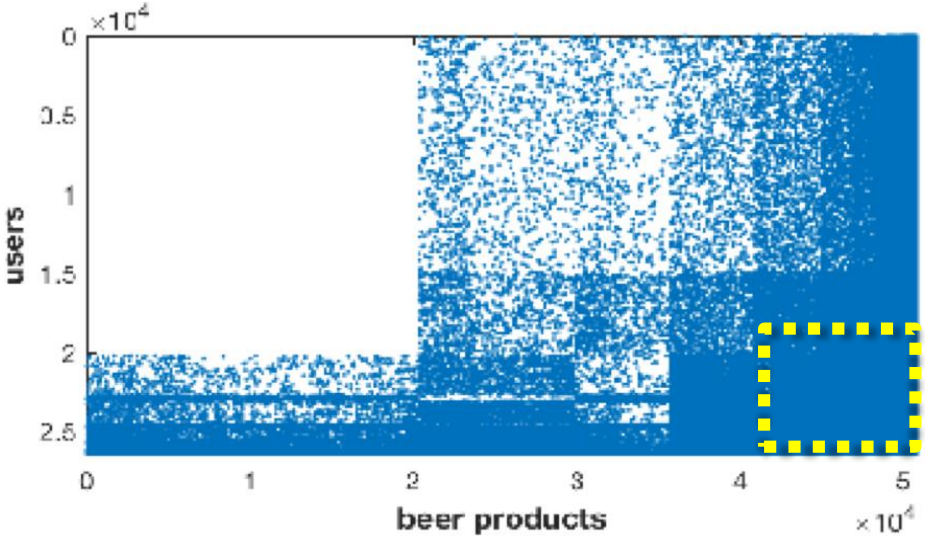
Fraudar: bounding graph fraud in the face of camouflage [B Hooi et al, KDD'16]

Challenge I: Hyperbolic community exists in real graphs



Communities found in YouTube friendship and Wikipedia articles [SNAP datasets]

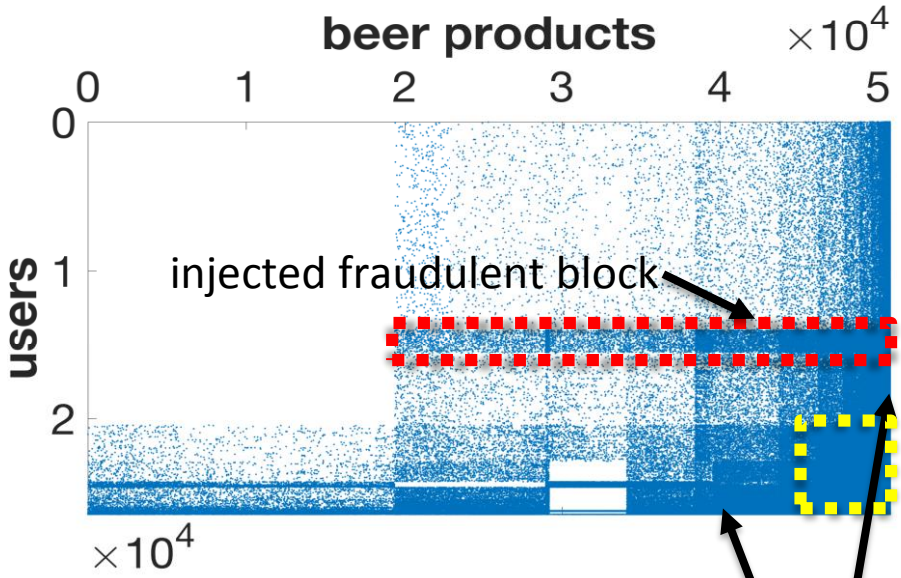
HyCoM Model: $i^{\alpha} \cdot j^{\alpha} > \tau$,
$0 < \tau < 1$ and $\alpha = -0.6 \sim -1.5$



Visualization of BeerAdvocate data

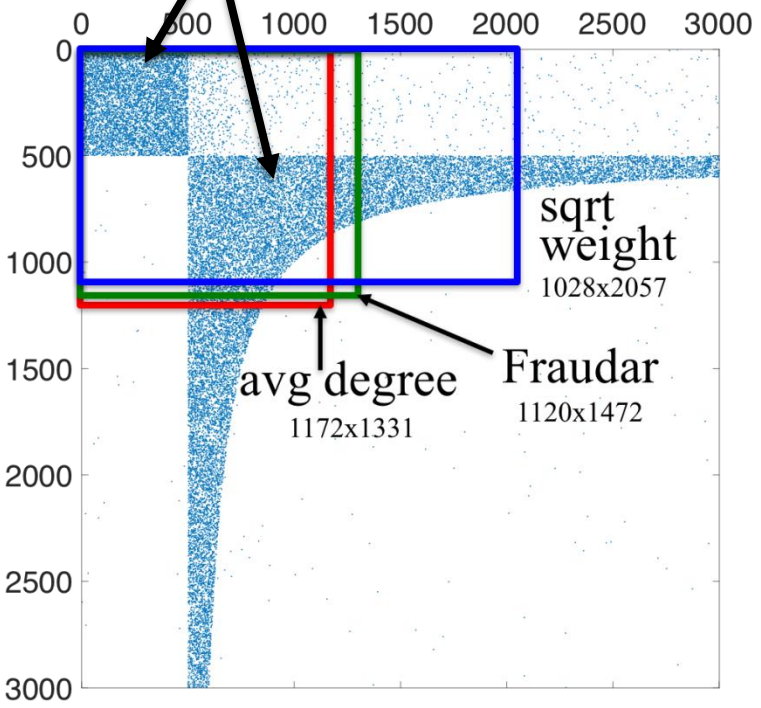
cross-association [D Chakrabarti et al, KDD'04]; Hyperbolic community detection [M Araujo et al, ECML-PKDD'14]; SNAP datasets: http://snap.stanford.edu/data/index.html

How can we avoid detecting the false positive hyperbolic block?



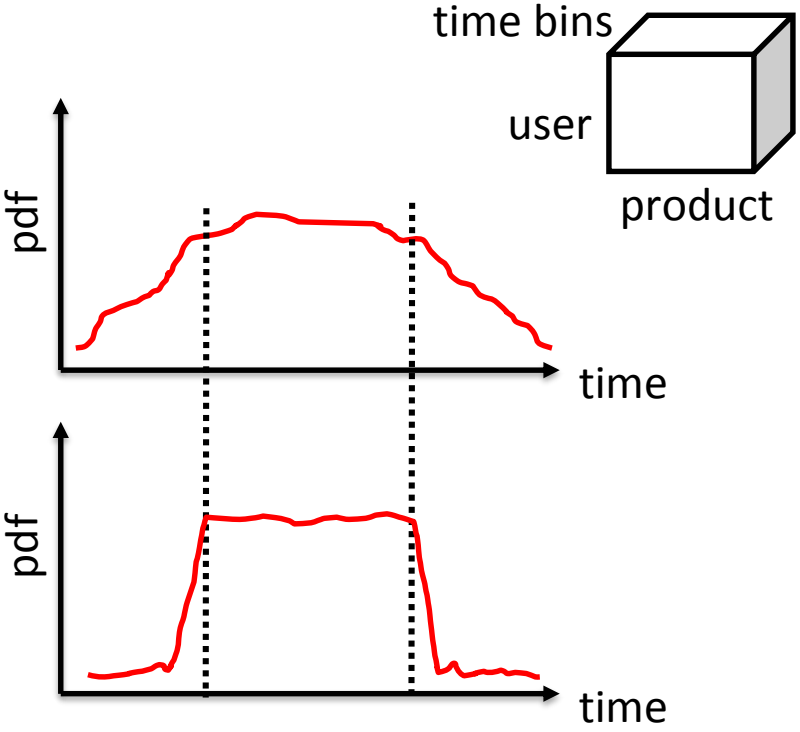
hyperbolic block (naturally formed)

Penalize sink nodes in both blocks



Synthetic data

Challenge II: Consider temporal information in fraud detection



Tensor-based methods (M-Zoom, D-Cube, CrossSpot) detect the two cases as the same density level in temporal dim.

Comparison with existing methods

	Fraudar	SpokEn	CopyCatch	CrossSpot	BP-based methods	M-Zoom/D-Cube	HoloScope
scalability	✓	✓	✓	✓	✓	✓	✓
camouflage	✓		?	✓		✓	✓
hy-community			?	?			✓
spike-aware			?				✓

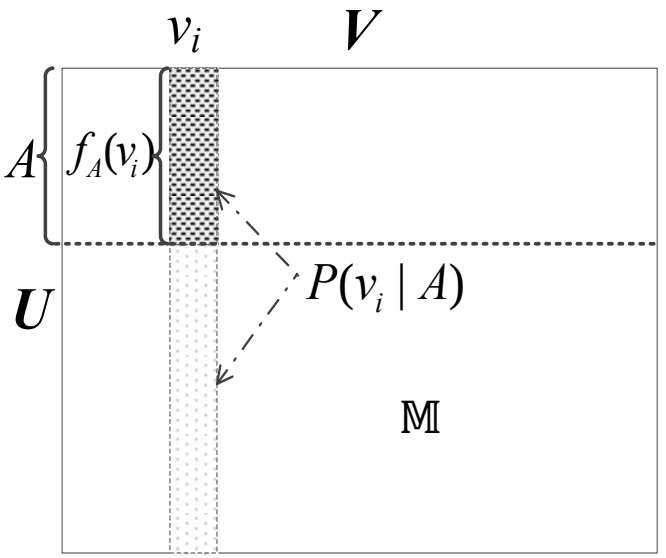
"hy-community" : avoid detecting the naturally-formed hyperbolic topology

Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
- Experiments
- Conclusion

Contrast suspiciousness in HoloScope

- $D(A,B) = \frac{\sum_{v_i \in B} f_A(v_i)}{|A|+|B|}$
  - $A \subset U, B \subset V$
  - $f_A(v_i) = \sum_{(u_j, v_i) \in E \wedge u_j \in A} \sigma_{ji} \cdot e_{ji}$, σ_{ji} is edge weight
- Contrast susp: $P(v_i \in B | A)$
  - the conditional likelihood



- Objective:

$$\max_A HS(A) := \mathbb{E}[D(A,B)]$$

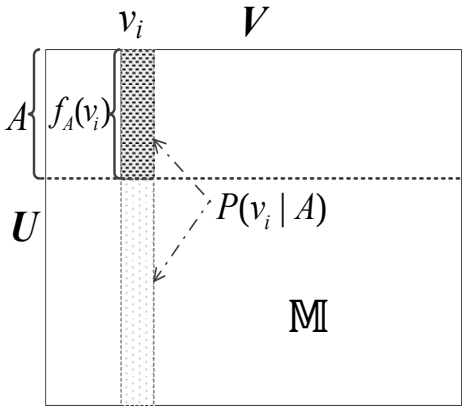
$$= \frac{1}{|A| + \sum_{k \in V} P(v_k|A)} \sum_{i \in V} f_A(v_i) P(v_i|A)$$

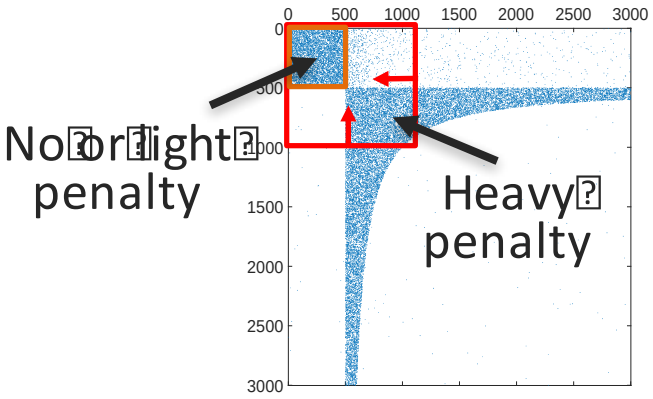
Detailed Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
  - **Topology-aware HS-α**
  - Temporal-spike aware
  - HS: make holistic use of signals
  - Scalable Algorithm
- Experiments
- Conclusion

Topology-aware (dense block) HS-α

- $P(v_i|A) = q(\alpha_i), \alpha_i = \frac{f_A(v_i)}{f_U(v_i)}$
  - Scaling fun: $q(x) = b^{x-1}, 0 \leq x \leq 1$ and constant $b > 1$
- users' susp score:
  - $S(u_j \in A) = \sum_{u_j v_i \in E} \sigma_{ji} e_{ji} \cdot P(v_i|A)$

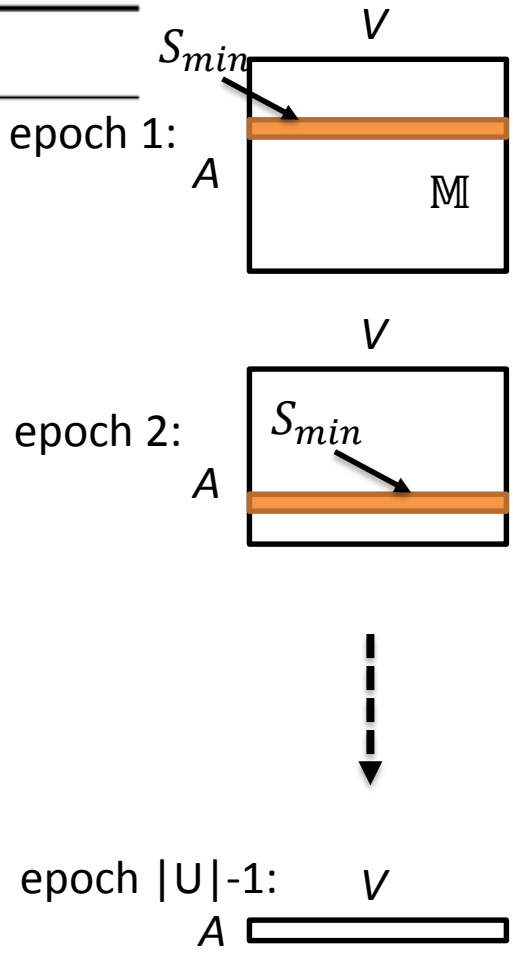




Algorithm HS-α considers topology

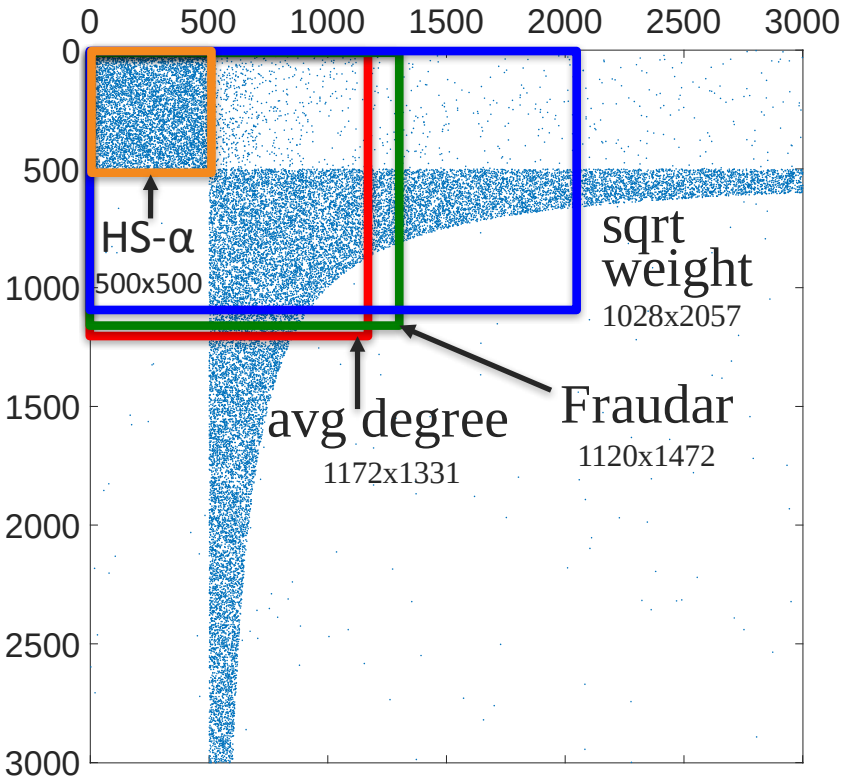
Algorithm 1 HS-α Algorithm.

Given adjacency matrix $\mathbb{M}$
Initialize:
    $A = U$
    $\mathcal{P}$= calculate contrast susp of all sink nodes given A
$\mathcal{S}$ = calculate susp scores of source nodes A.
MT = build priority tree of A with scores $\mathcal{S}$.
while A is not empty **do**
    u = pop the source node of the minimum score from MT.
    $A = A \setminus u$, delete u from A.
    Update $\mathcal{P}$ with respect to new source nodes A.
    Update MT with respect to new $\mathcal{P}$.
    Keep A^* that has the largest objective $HS(A^*)$
end while
return A^* and $P(v|A^*), v \in V$.



HS-α can shrink the detection box over hyperbolic community

- Synthetic data
  - Scaling fun: $q(\alpha_i) = 128^{\alpha_i - 1}$
  - b = 128

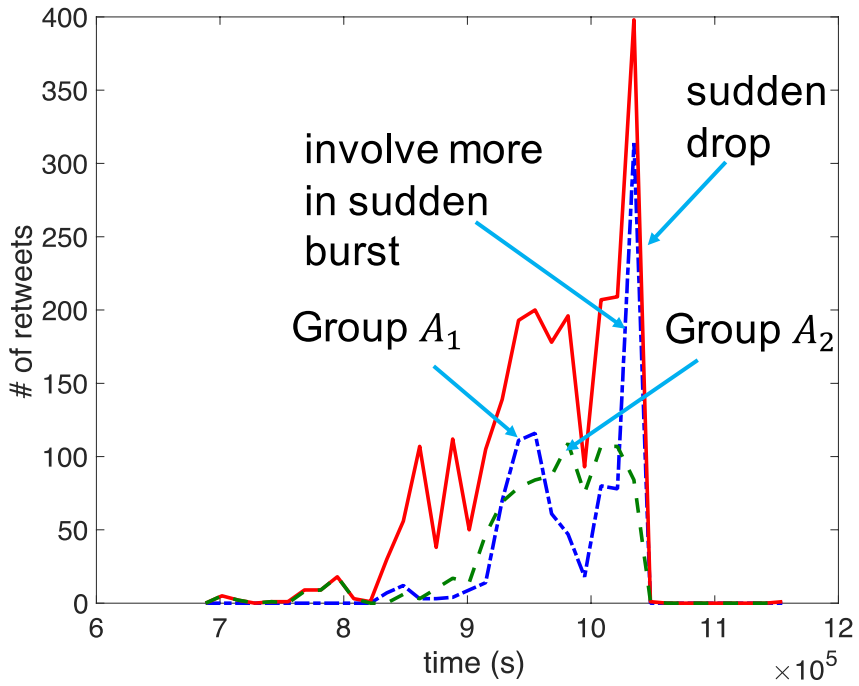


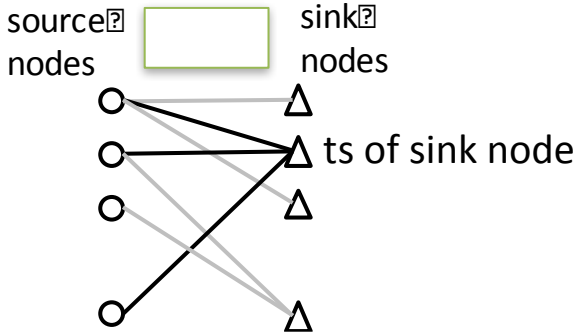
Detailed Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
  - Topology-aware HS-α
  - **Temporal-spike aware**
  - HS: make holistic use of signals
  - Scalable Algorithm
- Experiments
- Conclusion

Temporal spike: burst and drop are suspicious

- The histogram (time series) of a sink node
  - users retweet a message in Sina Weibo data.



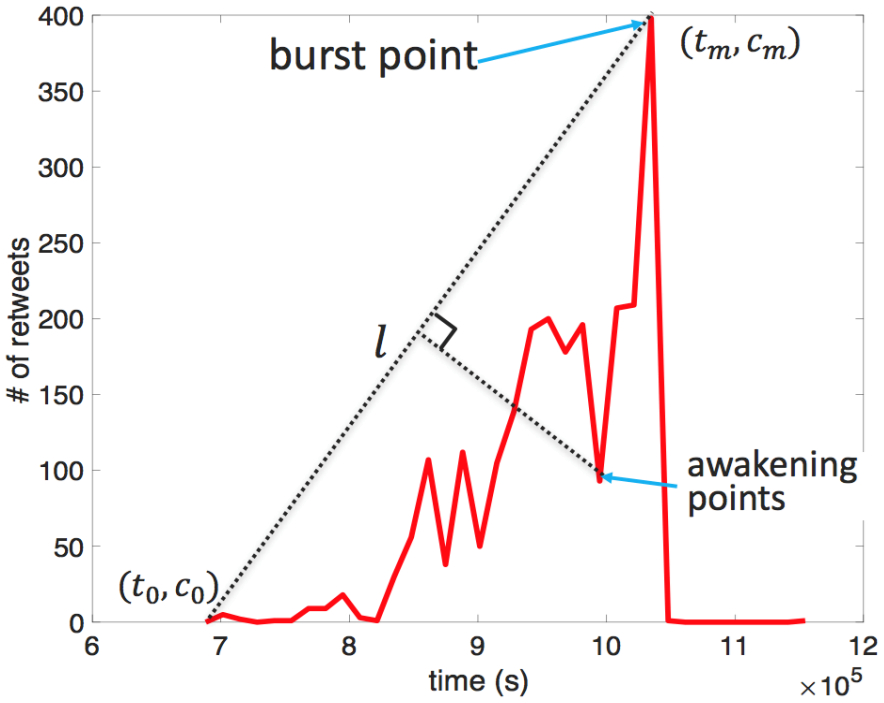


Group A_1 and A_2 has the same total # of retweets

A_1 is more suspicious than A_2

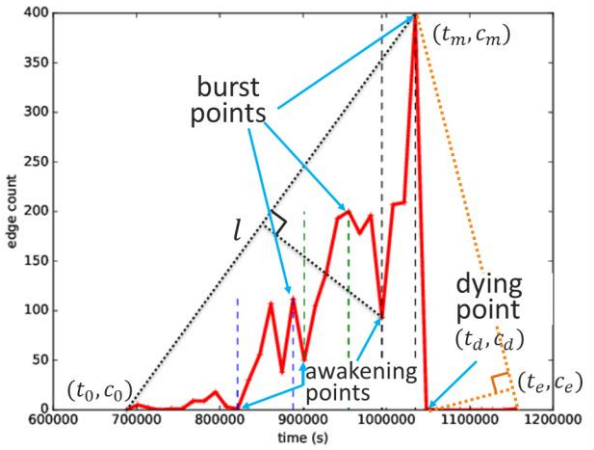
Detect spikes in time series of a sink node

- SB (Sleeping Beauty) defines burst and awakening point
- drop and dying points



awakening point: the point has the largest distance to l

$$t_a = \underset{(c,t)\in\mathcal{T}, t<t_m}{\arg\max} \frac{|(c_m - c_0)t - (t_m - t_0)c + t_m c_0 - c_m t_0|}{\sqrt{(c_m - c_0)^2 + (t_m - t_0)^2}}$$



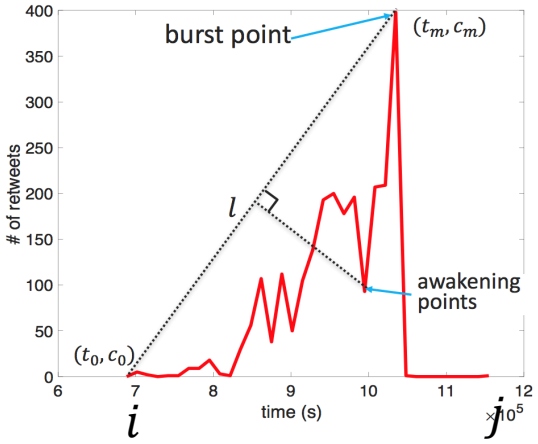
Detecting and identifying Sleeping Beauties in science [Ke et al, PNAS'15]

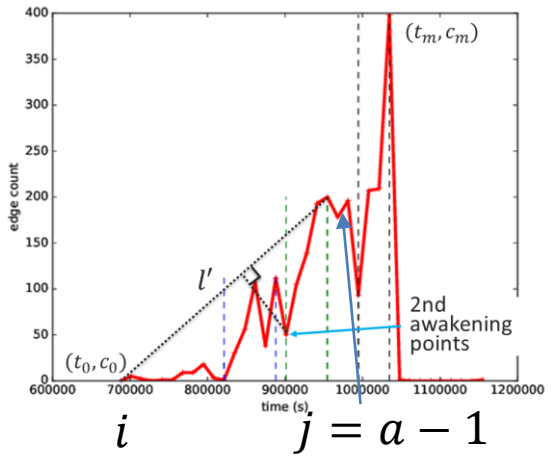
Multiburst algorithm

- Recursively detect burst and awakening points

Algorithm 2 *MultiBurst* algorithm.

Input Time series $\mathcal{T}$ of sink node v, beginning index i, end index j
Output A list of awakening-burst point pairs,
s_{am}: slope of the line passing through each point pair,
Δc: altitude difference of each point pair.
If $j - i < 2$ **then return**
(t_m, c_m) = point of maximum altitude between i and j.
(t_a, c_a) = the awakening point between i and j.
$\Delta c_{am} = c_m - c_a$, and $s_{am} = {}^{\Delta c_{am}}/_{(t_m - t_a)}$
Append $\{(t_a, c_a), (t_m, c_m)\}$, s_{am}, and Δc_{am} into the output.
$MultiBurst(\mathcal{T}, i, a-1)$
k = Find the first local min position from indices $m+1$ to j
$MultiBurst(\mathcal{T}, k, j)$

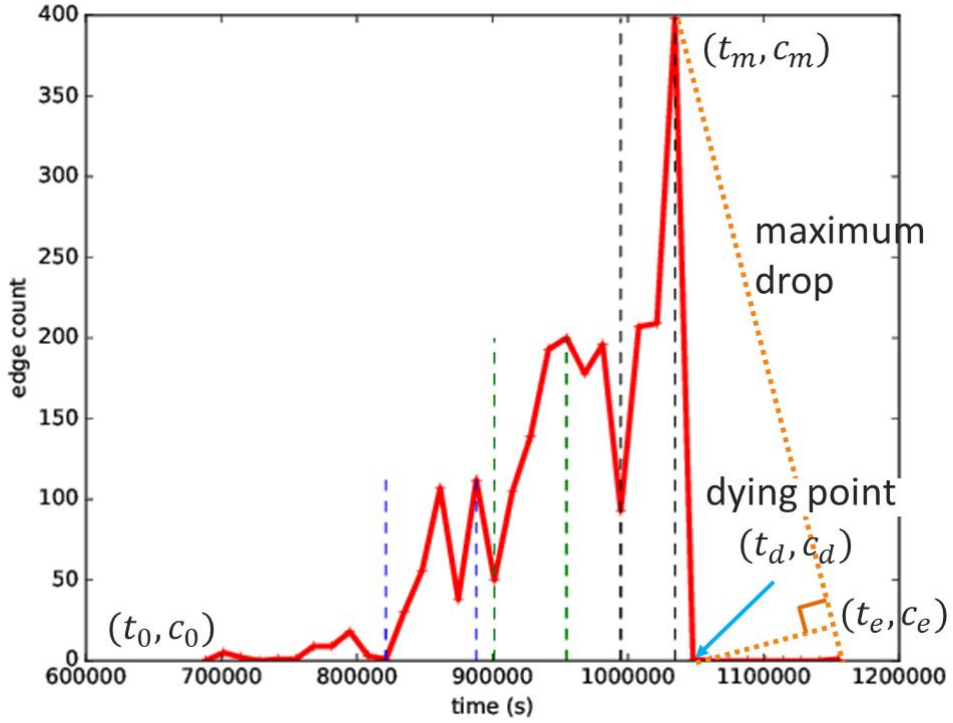




Find the maximum drop

- Recursively find the drop and return the maximum

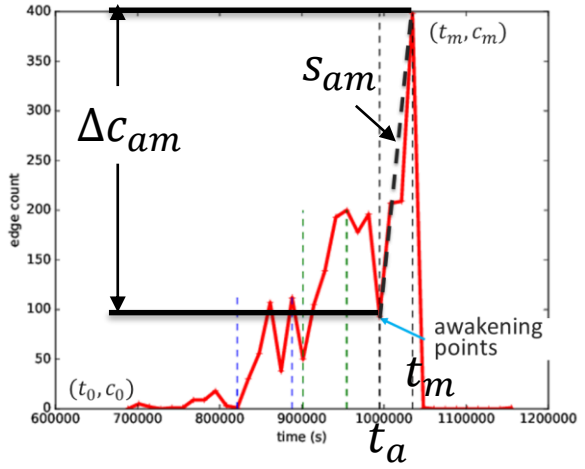
skip detail



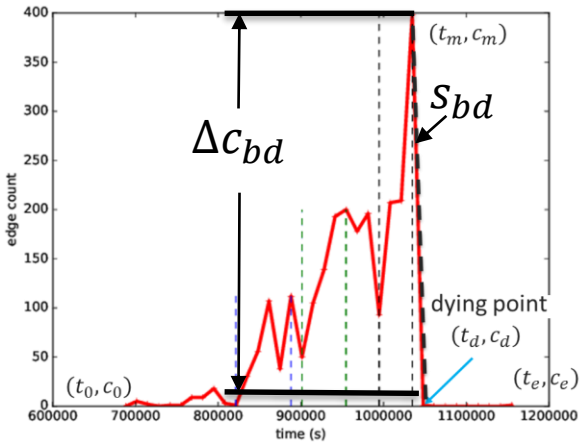
HoloScope considers time spikes

- multibust
  - $P(v_i|A) = q(\varphi_i),\ \varphi_i = \dfrac{\Phi(T_A)}{\Phi(T_U)}$

$$\Phi(T) = \sum_{(t_a, t_m)} \Delta c_{am} \cdot s_{am} \sum_{t \in T} \mathbf{1}(t \in [t_a, t_m])$$



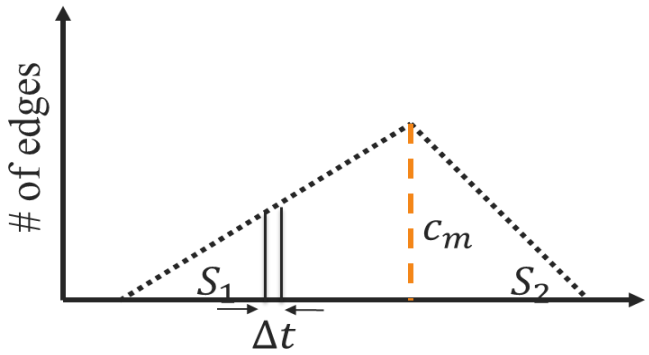
- sudden drop
  - $f_A(v_i) = \sum_j \sigma_{ji} e_{ji}$
  - $\sigma_{ji} = \Delta c_{bd} \cdot s_{bd}$



$$\max_A HS(A) := \mathbb{E}[D(A,B)] = \frac{1}{|A| + \sum_{k \in V} P(v_k|A)} \sum_{i \in V} f_A(v_i) P(v_i|A)$$

Time obstruction for fraudsters

- Theorem 1



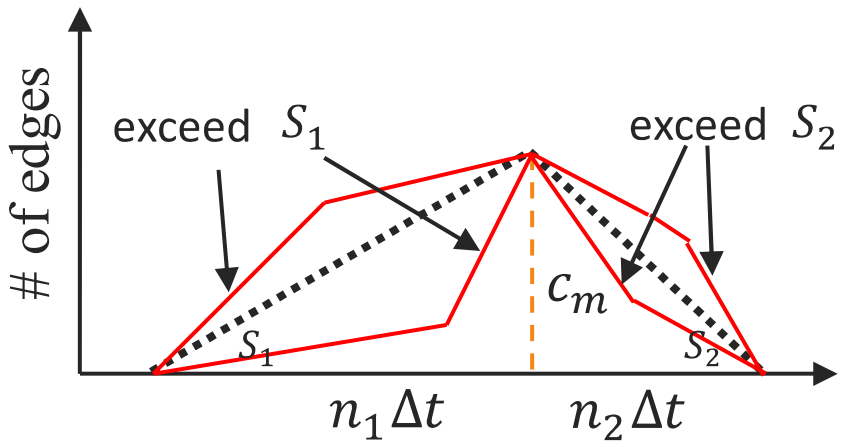
Let *N* be the number of edges that fraudsters want to create for an object.
If the fraudsters use time less than

$$\tau \geq \sqrt{\frac{2N\Delta t \cdot (S_1 + S_2)}{S_1 \cdot S_2}}$$

then they will be tracked by a suspicious burst or drop.

- Δt is the size of time bins,
- S_1 and S_2 are the slopes of normal rise and decline respectively

Proof



$$\frac{c_m}{n_1\Delta t} = S_1, \qquad \frac{c_m}{n_2\Delta t} = S_2, \qquad (n_1 + n_2)\cdot c_m = 2N'.$$

- n1 and n2 are # of time bins before and after the burst.
- *N*' is the total # of rating edges, and $N' \geq N$

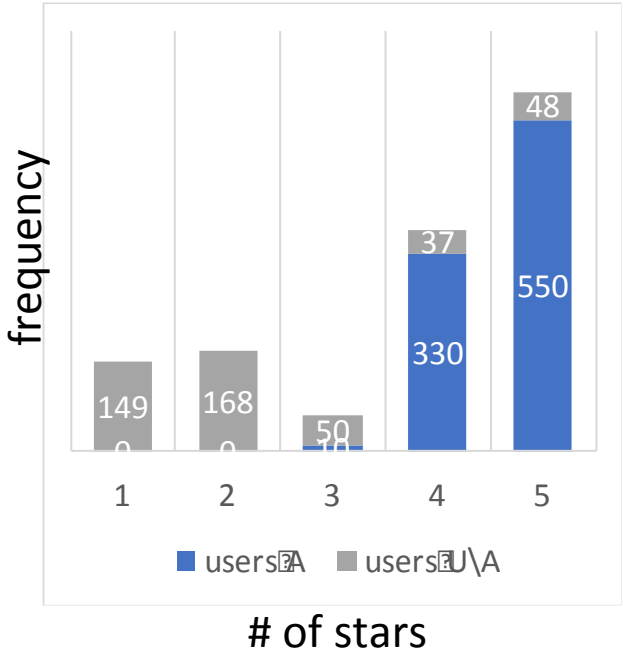
$$\tau = (n_1 + n_2)\Delta t = \sqrt{\frac{2N'\Delta t(S_1 + S_2)}{S_1 \cdot S_2}} \geq \sqrt{\frac{2N\Delta t(S_1 + S_2)}{S_1 \cdot S_2}}$$

Detailed Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
  - Topology-aware HS-α
  - Temporal-spike aware
  - **HS: make holistic use of signals**
  - Scalable Algorithm
- Experiments
- Conclusion

HS: make holistic use of signals

- Topology awareness: $\alpha_i = \frac{f_A(v_i)}{f_U(v_i)}$
- Temporal-spike awareness: $\varphi_i = \frac{\Phi(T_A)}{\Phi(T_U)}$
- Rating deviation: κ_i
  - κ_i=KL-divergence$(A, U \backslash A)$
  - $\kappa_i \leftarrow \kappa_i \cdot \min\{\frac{f_A(v_i)}{f_{U\backslash A}(v_i)}, \frac{f_{U\backslash A}(v_i)}{f_A(v_i)}\}$
- Contrast susp of HS
  - $P(v_i|A) = \boldsymbol{q}(\alpha_i)\boldsymbol{q}(\varphi_i)\boldsymbol{q}(\kappa_i) = b^{\alpha_i+\varphi_i+\kappa_i-3}$
  - "joint probability"



$$\max_A HS(A) := \mathbb{E}[D(A,B)] = \frac{1}{|A| + \sum_{k \in V} P(v_k|A)} \sum_{i \in V} f_A(v_i) P(v_i|A)$$

Using the same algorithm framework

- Find burst and drop points of each sink node
  - cost $O(d_v)$, total cost $O(|E|)$
- Use framework of HS-α algorithm

Algorithm 3 HS algorithm (unscalable).

Given bipartite multigraph $\mathcal{G}(U, V, E)$,
initial source nodes $A_0 \subset U$.
Initialize:
$A = A_0$
$\mathcal{P}$= calculate contrast suspiciousness given A_0
$\mathcal{S}$ = calculate suspiciousness scores of source nodes A.
MT = build priority tree of A with scores $\mathcal{S}$. ← $O(m_0 \log m_0), m_0 = |A_0|$
while A is not empty **do**
u = pop the source node of the minimum score from MT.
$A = A \setminus u$, delete u from A.
Update $\mathcal{P}$ with respect to new source nodes A. ← $O(d_u \cdot |A|)$
Update MT with respect to new $\mathcal{P}$. ← $O(d_u \cdot |A| \cdot \log m_0)$
Keep A^* that has the largest objective $HS(A^*)$
end while
return A^* and $P(v|A^*)$, $v \in V$.

Time complexity

Algorithm 3 HS algorithm (unscalable).

Given bipartite multigraph $\mathcal{G}(U, V, E)$,
initial source nodes $A_0 \subset U$.
Initialize:
$A = A_0$
$\mathcal{P}$= calculate contrast suspiciousness given A_0
$\mathcal{S}$ = calculate suspiciousness scores of source nodes A.
MT = build priority tree of A with scores $\mathcal{S}$. ← $O(m_0 \log m_0), m_0 = |A_0|$
while A is not empty **do**
u = pop the source node of the minimum score from MT.
$A = A \setminus u$, delete u from A.
Update $\mathcal{P}$ with respect to new source nodes A. ← $O(d_u \cdot |A|)$
Update MT with respect to new $\mathcal{P}$. ← $O(d_u \cdot |A| \cdot \log m_0)$
Keep A^* that has the largest objective $HS(A^*)$
end while
return A^* and $P(v|A^*), v \in V$.

- The time complexity is
  - $\sum_{j=2,\cdots,m_0} O(d_j \cdot (j-1) \cdot \log m_0) = O(m_0 |E_0| \log m_0)$
  - When $A_0 = U$, it is $O(|U||E| \log |U|)$

Slow! Super quadratic # of nodes

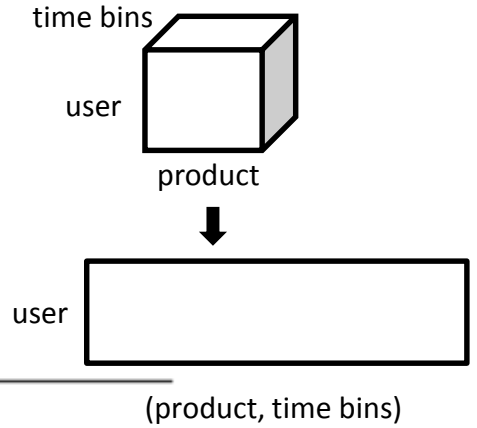
Scalable HS algorithm

- Main idea: feed small groups of users $\tilde{U}$ into *GreedyShaving* Procedure (previous HS alg.)

Algorithm 4 *FastGreedy* Algorithm for Fraud detection.

Given bipartite multigraph $\mathcal{G}(U, V, E)$.
$\mathbb{L}$ = get first several left singular vectors
for all $L^{(k)} \in \mathbb{L}$ **do**
  Rank source nodes U decreasingly on $L^{(k)}$
  $\tilde{U}^{(k)}$ = truncate $u \in U$ when $L_u^{(k)} \leq \frac{1}{\sqrt{|U|}}$
  GreedyShaving with initial $\tilde{U}^{(k)}$.
end for
return the best A^* with maximized objective $HS(A^*)$,
  and the rank of $v \in V$ by $f_{A^*}(v) \cdot P(v|A^*)$.

To consider temporal and #star information, we matricize tenor into a matrix

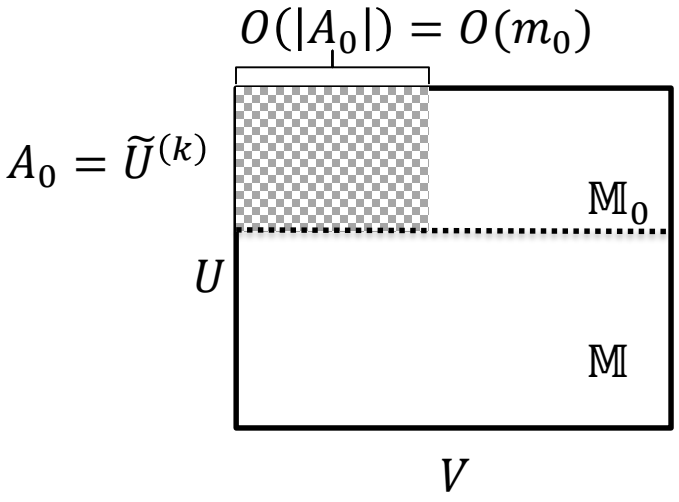


Scalable HS alg is sub-quadratic # of nodes

- Theorem 2 (algorithm complexity)

> Given $|V| = O(|U|)$ and $|E| = O(|U|^{\epsilon_0})$,
> the time complexity of *FastGreedy* is subquadratic,
> $\boldsymbol{o(|U|^2)}$ in little-*o* notation,
> if $|\widetilde{U}^{(k)}| \leq |U|^{1/\epsilon}$, where $\epsilon > \max\{1.5, \frac{2}{3-\epsilon_0}\}$

Proof



- The total number of edges in $\mathbb{M}_0$ is

$$O(|E_0|) = O(m_0^2 + \frac{m_0 \cdot |E|}{|U|}) = O(|U|^{2/\epsilon} + |U|^{1/\epsilon - 1}|E|)$$

- So the HS algorithm complexity is

$$O(m_0|E_0| \log m_0) = O((|U|^{3/\epsilon} + |U|^{2/\epsilon - 1 + \epsilon_0}) \log |U|)$$

$|\widetilde{U}^{(k)}| \leq |U|^{1/\epsilon}$

- Therefore, if $\epsilon > \max\{1.5, \frac{2}{3-\epsilon_0}\}$, the complexity is subqudratic $o(|U|^2)$. ($\epsilon \leq 2$?)
- In real life graph, if $\epsilon_0 \leq 1.6$, so we can limit $|\widetilde{U}^{(k)}| \leq |U|^{1/1.6}$

$|E| = O(|U|^{\epsilon_0})$

Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
- Experiments
- Conclusion

Data sets

Table 1: Data Statistics

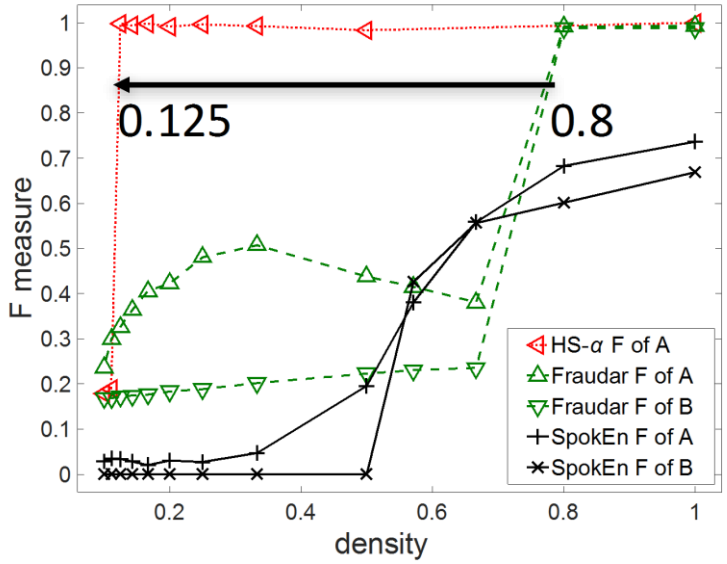
Data Name	#nodes	#edges	time span
BeerAdvocate	26.5K x 50.8K	1.07M	Jan 08 - Nov 11
Yelp	686K x 85.3K	2.68M	Oct 04 - Jul 16
Amazon Phone & Acc	2.26M x 329K	3.45M	Jan 07 - Jul 14
Amazon Electronics	4.20M x 476K	7.82M	Dec 98 - Jul 14
Amazon Grocery	763K x 165K	1.29M	Jan 07 - Jul 14
Amazon mix category	1.08M x 726K	2.72M	Jan 04 - Jun 06

Data sets are published by [J McAuley and J Leskovec, RecSys'13] [J McAuley and J Leskovec, WWW'13] [A Mukherjee et al, WWW'12]

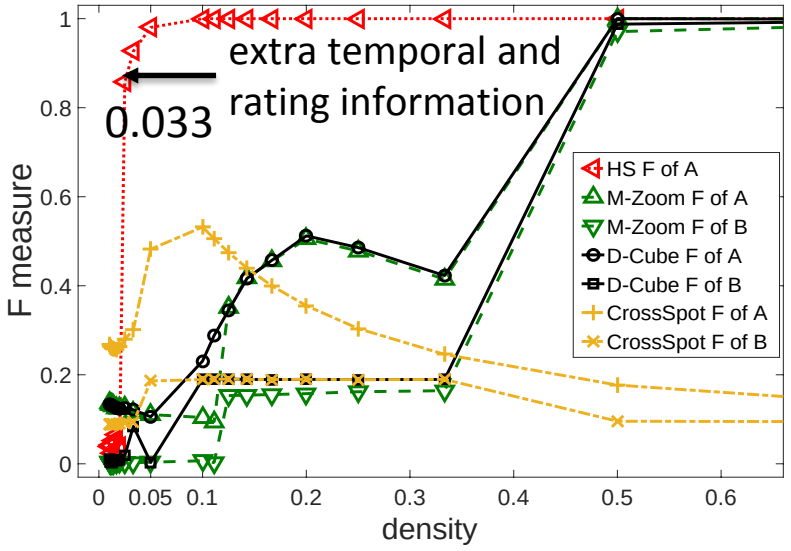
Performance on injected labels

- Mimic fraudsters to inject edges, time stamps and #stars, with different fraudulent density

BeerAdvocate Data



HS-α consider only topology (density)

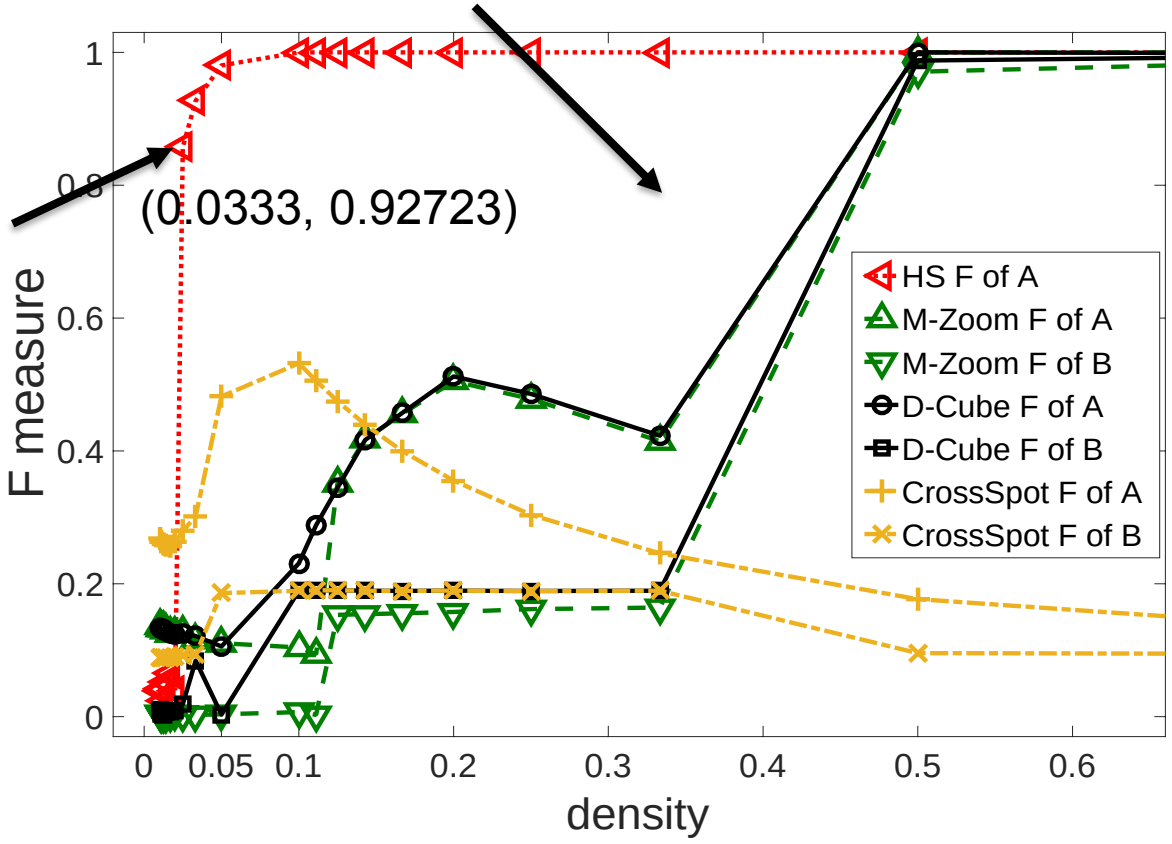


HS consider all signals

We use two quantitative metrics for comparison

1. “auc”: the area under the curve of the accuracy curve

2. **lowest detection density (L.D.D.)**: the density that a method can detect in high accuracy (”≥ 90%”).



The fraudulent density ranges from 1.0 to 0.01 for testing

Performance on injected labels by mimicking

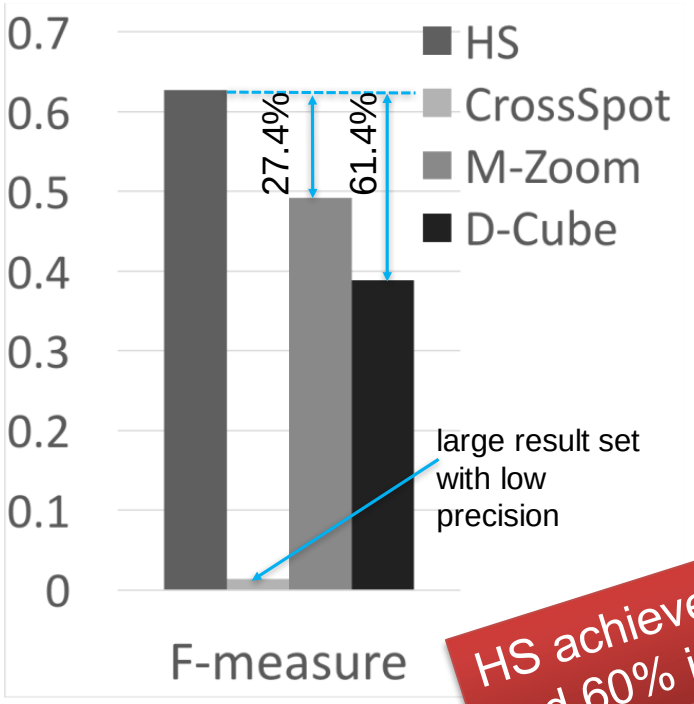
Data Name	metrics*	source nodes				sink nodes			
		M-Zoom	D-Cube	CrossSpot	HS	M-Zoom	D-Cube	CrossSpot	HS
BeerAdvocate	auc	0.7280	0.7353	0.2259	**0.9758**	0.6221	0.6454	0.1295	**0.9945**
	F≥90%	0.5000	0.5000	–	**0.0333**	0.5000	0.5000	–	**0.0333**
Yelp	auc	0.9019	0.9137	0.9916	**0.9925**	0.9709	0.8863	0.0415	**0.9950**
	F≥90%	0.2500	0.2000	0.0200	**0.0143**	0.0250	1.0000	–	**0.0100**
Amazon Phone & Acc	auc	0.9246	0.8042	0.0169	**0.9691**	0.9279	0.8810	0.0515	**0.9823**
	F≥90%	0.1667	0.5000	–	**0.0200**†	0.1429	0.1000	–	**0.0200**†
Amazon Electronics	auc	0.9141	0.9117	0.0009	**0.9250**	0.9142	0.7868	0.0301	**0.9385**
	F≥90%	0.2000	0.1250	–	**0.1000**	**0.1000**	0.5000	–	0.1250
Amazon Grocery	auc	0.8998	0.8428	0.0058	**0.9250**	0.8756	0.8241	0.0200	**0.9621**
	F≥90%	0.1667	0.5000	–	**0.1000**	0.1250	0.2500	–	**0.1000**
Amazon mix category	auc	0.9001	0.8490	0.5747	**0.9922**	0.9937	0.9346	0.0157	**0.9950**
	F≥90%	0.2500	0.5000	0.2000†	**0.0167**	**0.0100**	0.2000	–	**0.0100**

* the performance is very stable when *b* larger than 32.

- HS achieved the best auc, and even reached the testing *upper bound* (0.9950) in two cases
- HS has L.D.D. as small as 200/14000=0.0143 on source nodes, the minimum test density 0.01 on sink nodes.

Performance on real labels from online system

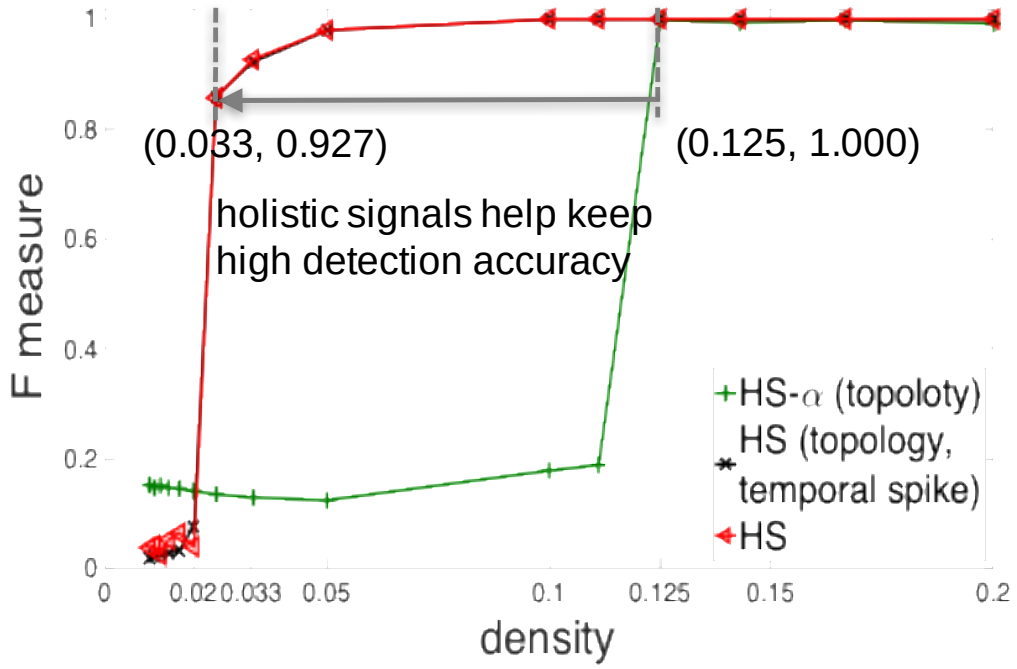
- Sina Weibo is a microblog and Twitter-like website
  - 2.75 M users, 8.08 M messages, and 50.1 M edges in our data of Dec 2013



1. Union of results from all methods
2. Crawl from the website to see if they are removed from the website
3. Result in 3957 labeled users and 1615 labeled messages for susp

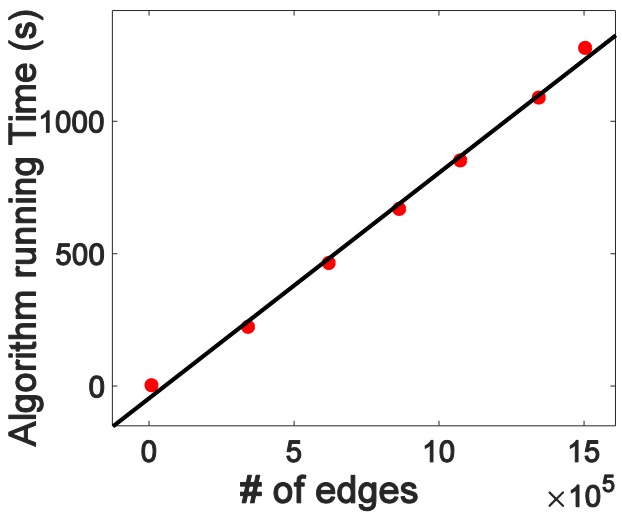
HS achieved more than 30% and 60% improvement

Effective to combine multiple signals

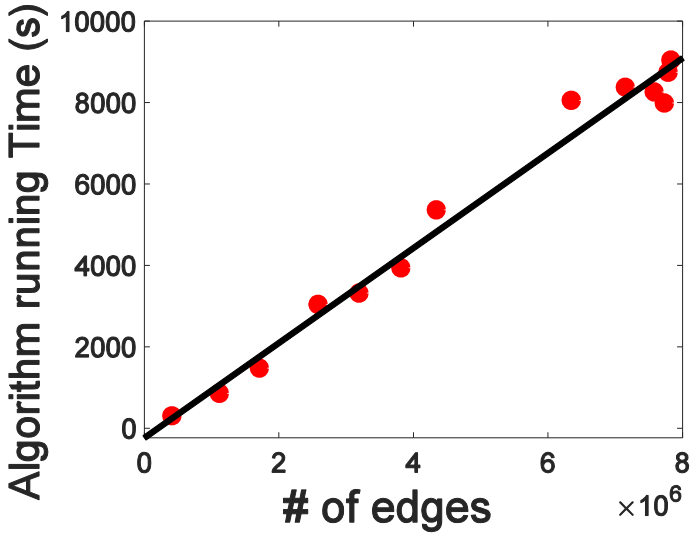


HS (topology, temporal spike) is close to the best, missing 30 of 6,000 fraudsters to recall

Scalability



BeerAdvocate dataset



Amazon Electronics dataset

HoloScope runs in near-linear time of # of edges

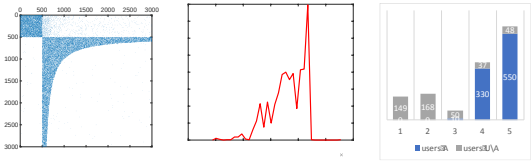
Outline

- Background and Problem
- Graph-based fraud detection
- HoloScope Algorithm
- Experiments
- Conclusion

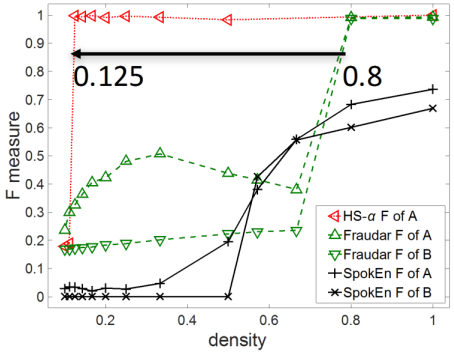
Conclusion and taking away

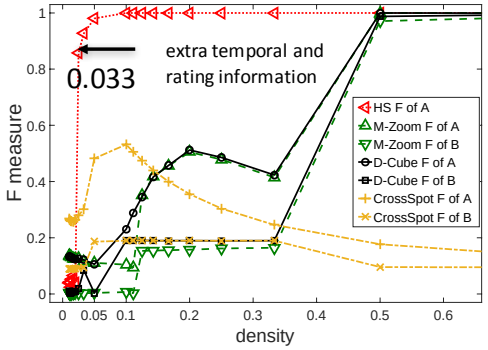
- HoloScope:
  - Fraud detection on (user, object, timstamp, #stars)
- Unification of signals
  - topology, temporal spikes, and rating deviation
- Theoretical analysis of fraudsters' obstruction
- Effectiveness
- Scalability

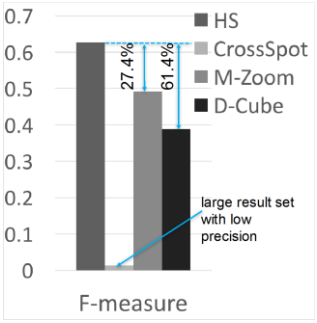
topology spikes rating

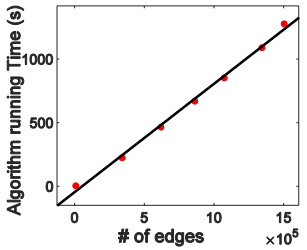


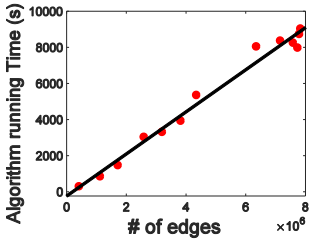
$$\tau \geq \sqrt{\frac{2N\Delta t \cdot (s_1 + s_2)}{s_1 \cdot s_2}}$$









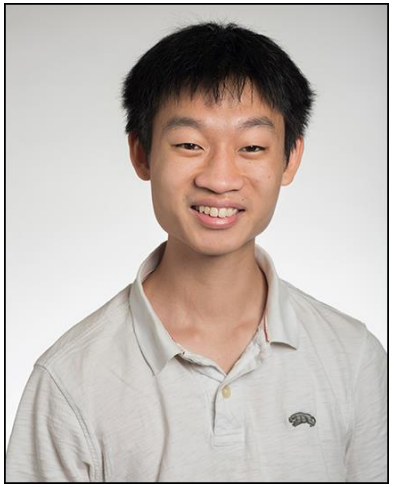
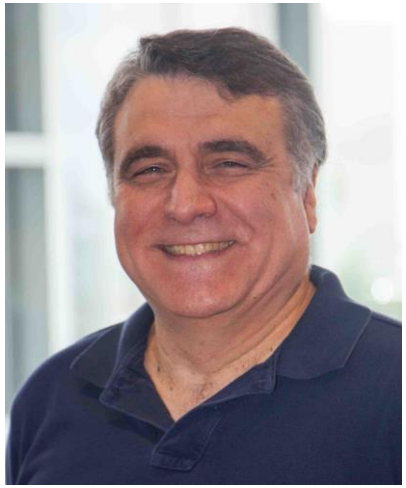


More information about HoloScope

- CIKM 2017
- KDD MLG workshop version:
  - http://www.mlgworkshop.org/2017/paper/MLG2017_paper_3.pdf
- Source code
  - https://github.com/shenghua-liu/HoloScope

Reference

- **[Charikar M, 2000]** Charikar, Moses. "Greedy approximation algorithms for finding dense components in a graph." *International Workshop on Approximation Algorithms for Combinatorial Optimization*, 2000.
- **[Asahiro et al, SWAT'96]** Asahiro, Yuichi, et al. "Greedily finding a dense subgraph." *Algorithm Theory—SWAT'96* (1996): 136-148.
- **[B Hooi et al, KDD'16]** Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos
- Faloutsos. 2016. Fraudar: bounding graph fraud in the face of camouflage. KDD 2016
- **[M Araujo et al, ECML-PKDD'14]** Miguel Araujo, Stephan Gu¨nnemann, Gonzalo Mateos, and Christos Faloutsos. Beyond blocks: Hyperbolic community detection. ECML-PKDD, 2014. 50–65.
- **[M-Zoom]** Kijung Shin, Bryan Hooi, and Christos Faloutsos. M-Zoom: Fast Dense- Block Detection in Tensors with ality Guarantees. ECML-PKDD. 2016, 264–280.
- **[D-Cube]** Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. D-Cube: Dense-Block Detection in Terabyte-Scale Tensors. WSDM '17. 2017.
- **[CrossSpot]** Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. A general suspiciousness metric for dense blocks in multimodal data. ICDM, 2015, 781– 786.
- **[CopyCatch]** Alex Beutel, Wanhong Xu, Venkatesan Guruswami, Christopher Palow, and Christos Faloutsos. Copycatch: stopping group attacks by spotting lockstep behavior in social networks, WWW 2013. 119–130.
- **[SpokEn]** B Aditya Prakash, Mukund Seshadri, Ashwin Sridharan, Sridhar Machiraju, and Christos Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. PAKDD 2010, 290–295.
- **[Ke et al, PNAS'15]** Qing Ke, Emilio Ferrara, Filippo Radicchi, and Alessandro Flammini. Detecting and identifying Sleeping Beauties in science. PNAS, 112, 24 (2015), 7426–7431.

Questions and Answers

THANK YOU