Unique Experimental Algorithms for National-Security Applications

Cynthia A. Phillips (Sandia National Laboratories)

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Government Research/Applications

Priorities

- Public Safety
- Regulation compliance
- Situational awareness, information gathering (e.g. military)
- Protection of national assets (infrastructure, cyber)
- Efficient resource allocation
  - Budget constraints
  - Especially when a company (e.g. utility) will implement

Government Research/Applications

- Implementations: It must work
- Variety of platforms
  - High-performance computers to battery-powered systems
- High Confidence
  - Exact solutions
  - Mathematical proofs
  - Experimental evaluation of heuristics
    - Benchmarking
  - Uncertainty quantification
  - Validation

Three Stories

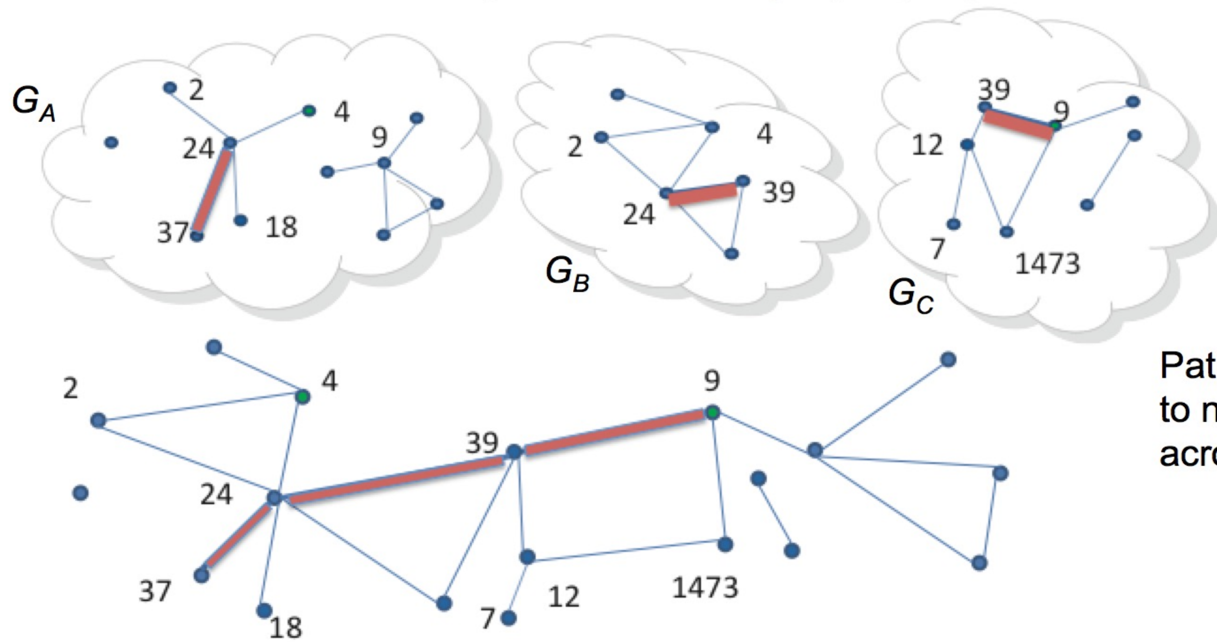
- Making social-network data sets more human
- Correctness of implementation: history-independent data sets
- When beautiful, simple, theoretically good algorithms are hard to implement: a special case of randomized rounding.

Origins: Distributed Graph Analytics

Alice and Bob (or more) independently create social graphs G_A and G_B.

Goal: Cooperate to compute algorithms over G_A union G_B with limited sharing: $O(\log^k n)$ total communication for size n graphs, constant k

Motivation: National security: "connect the dots" for counterterrorism

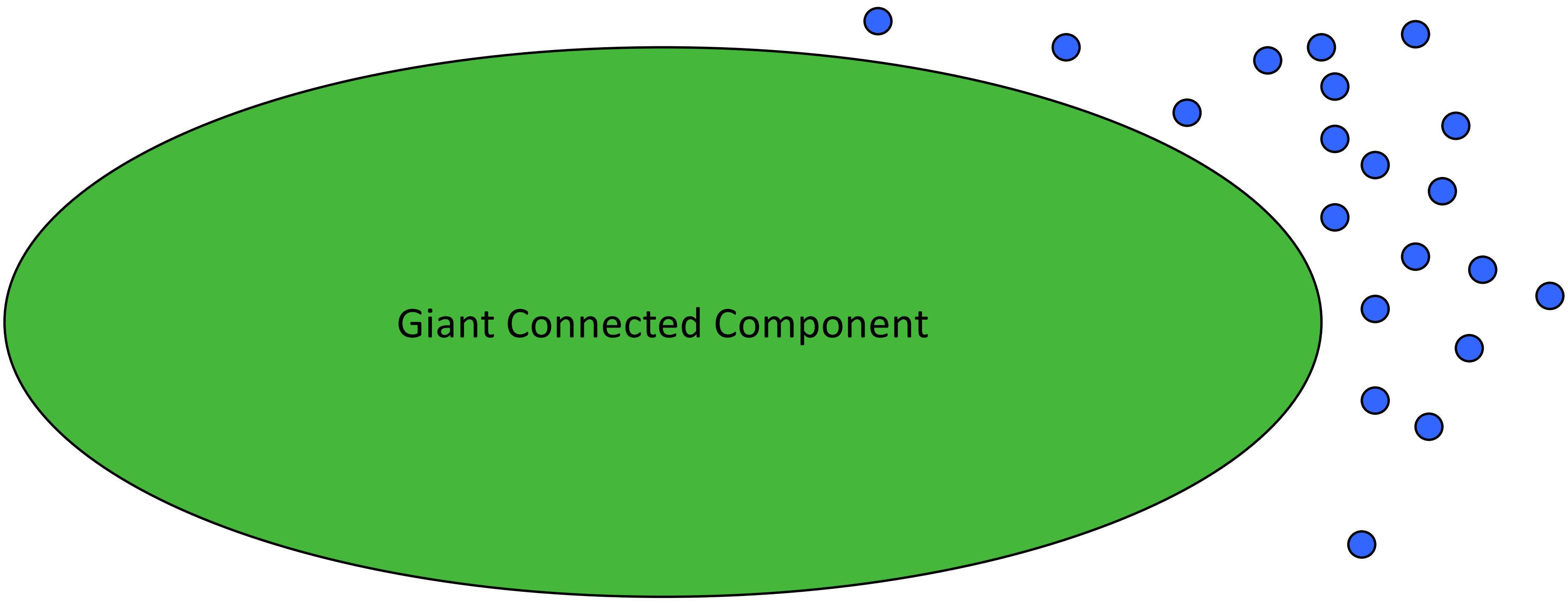


s-t connectivity
For 3 centers

Path from node s=37 to node t=9 is split across all three centers

Exploiting Graph Structure

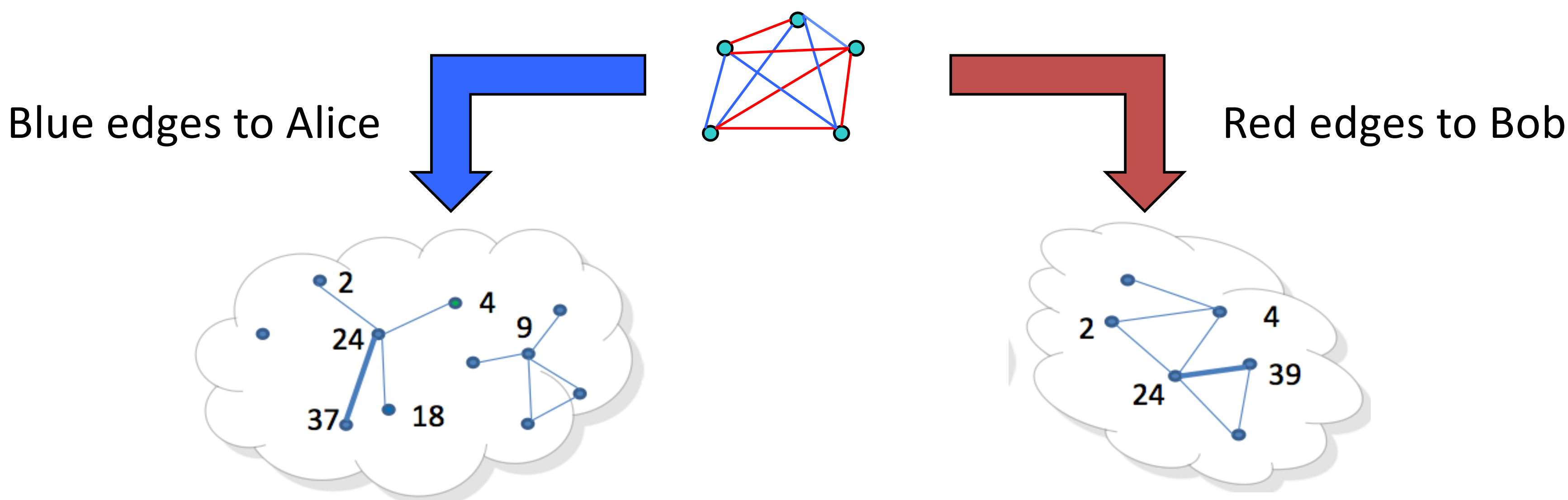
- Nodes are people, so exploit structure of social networks



- Past success: $O(\log^2 n)$-bit communication for s-t connectivity
  - Exploits giant component structure
  - Overcomes polynomial lower bounds for general graphs

Exploiting Social Network Structure

- Next step: planted clique (dense subgraph anomaly detection)
  - Structural conjectures based on evolutionary psychology
  - Provably correct algorithm
- Experimental validation on some real networks failed!



Human vs Automated

- Networks like Twitter contain a vast amount of non-human behavior
  - You can buy 500 followers for $5 US
  - Economic incentives to manipulate connections
- For our intended applications, the network owners (law-enforcement agencies) will have human-only networks
  - Networks are not public where entities can sign up
  - No cleaning problem
- We have no real data from law enforcement

Some Test Network Desired Properties

- Nodes are humans
- Edges plausibly represent a social bond
  - Even better if the relationship requires time/effort
- Large size (millions/billions of nodes/edges)
- Network is reasonably complete
  - Not an ego-network

Not too many publicly available social networks have all these.

Human vs Automated

Goal: Clean (enough) non-human behavior to test our algorithms

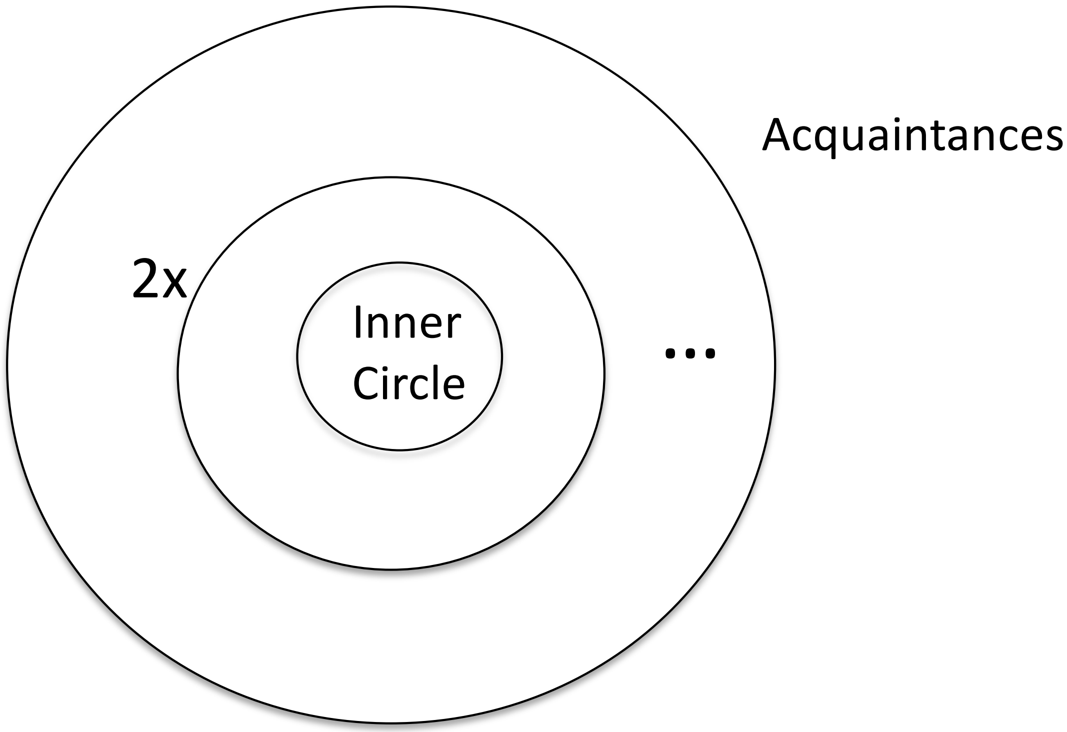
- Limitation: we have only topology
- An idea: Real human relationships require attention
  - Attention can be divided
  - Total attention, time of day, etc, is limited
- Nodes that show too many "strong" connections may not be human.
  - This includes humans, such as celebrities, who have a group of others manage their social media accounts.

- We'll give a method, then consider
  - Is it (plausibly) correct?
  - Should we care?

Varying Strength of Ties

- People “know” about 1500 others by face/name
- Hierarchy of strength

R. Dunbar, Social cognition on the internet: testing constraints on social network Size, Philosophical Transactions of the Royal Society B, Biological Sciences,367(1599):2192-2201, 2012

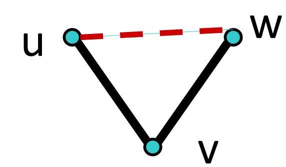


Bounded number of strong human interactions even with social media (Dunbar 2012)

Triangle Significance

- Strong triadic closure (Easley, Kleinberg): two strong edges in a wedge implies (at least weak) closure.

– Reasons: opportunity, trust, social stress



– Converse of strong triadic closure: not (both edges strong) implies coincidental closures

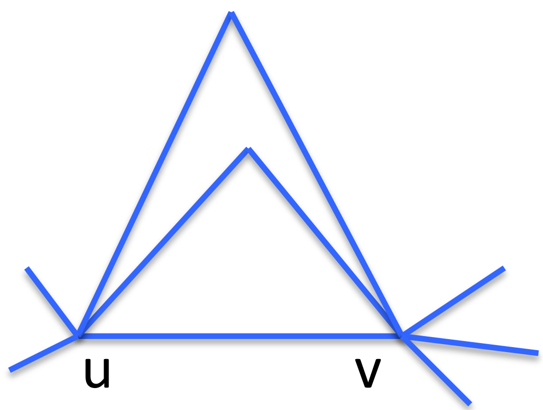
  - experimental evidence: Kossinets, Watts 2006

"Communities have triangles"

Edge strength

- A notion somewhat like Easley and Kleinberg 2010, and Berry et al., 2011

$$s(u,v) = \frac{2 * \#\ \text{triangles on}(u,v)}{d_u + d_v - 2}$$

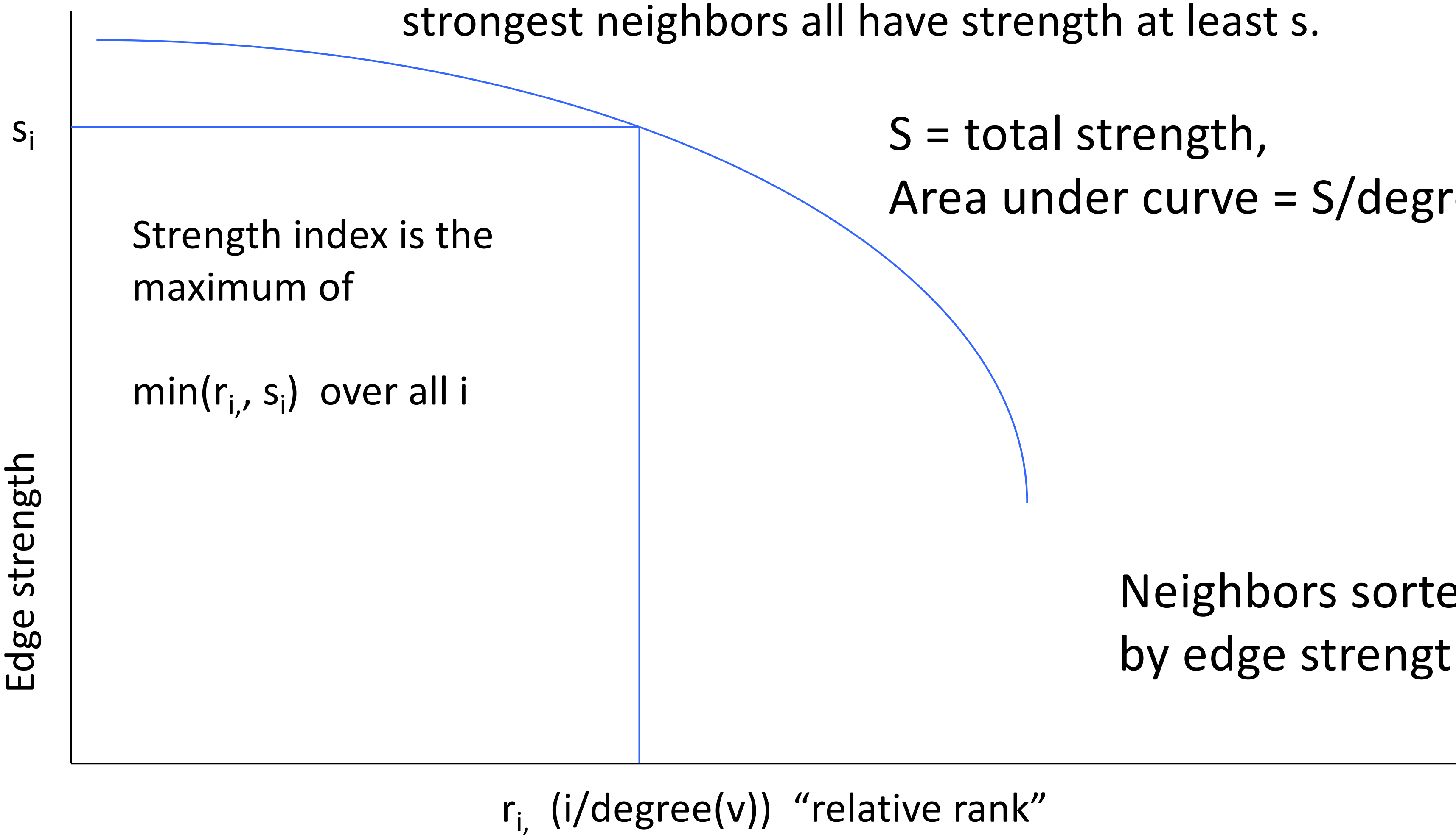


$$s(u,v) = \frac{2*2}{5+6-2} = \frac{4}{9}$$

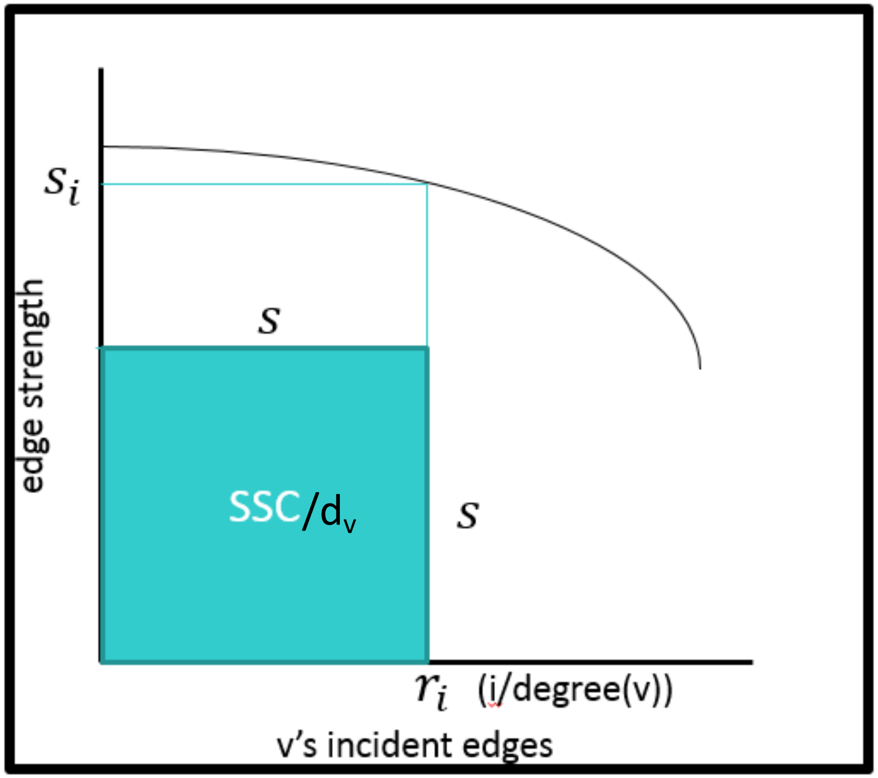
- Assumption: Total strength of edges on a vertex has a constant bound D_G (network-dependent)
  - Edge strength a continuum, not just strong/weak

Strength-index for a vertex

A strength index of s means that an s-fraction of the strongest neighbors all have strength at least s.



Strength-Index Property



Dunbar-like constant = D,
S = Sum of strengths<=s

Then: $D \geq S \geq s^2 * \text{ degree}$

$$s \leq \sqrt{\frac{D}{d}}$$

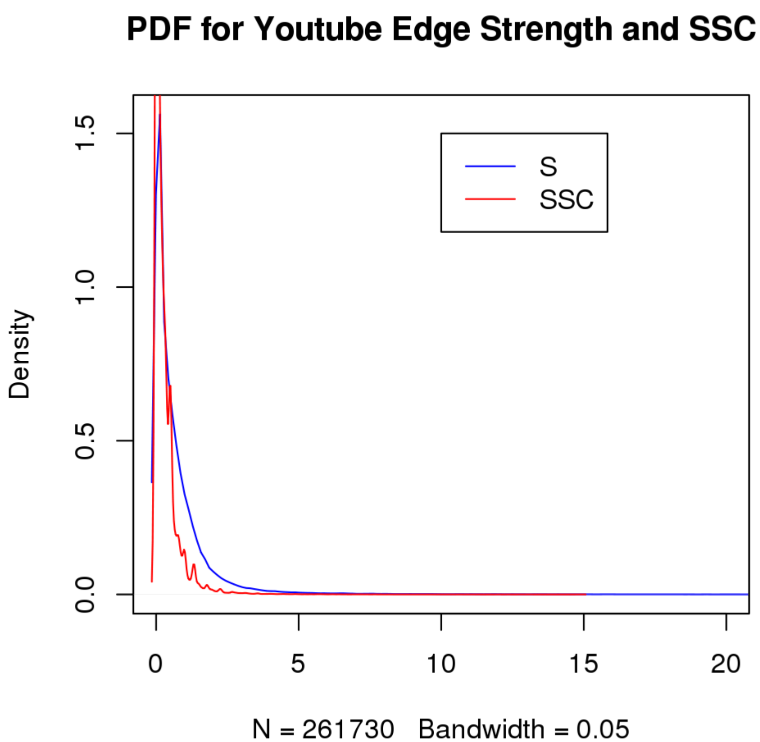
s = s-index
D = Dunbar-like constant
d = degree
SSC = $s^2 d_v$

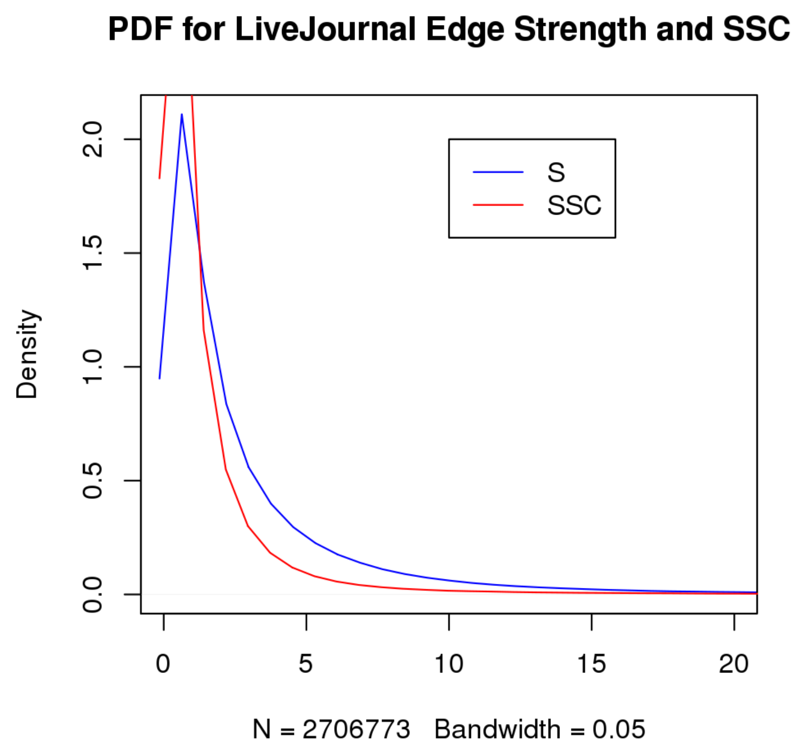
SSC = "Symmetric strength component"

Most important edges
Free from tail effects

SSC and total strength distributions

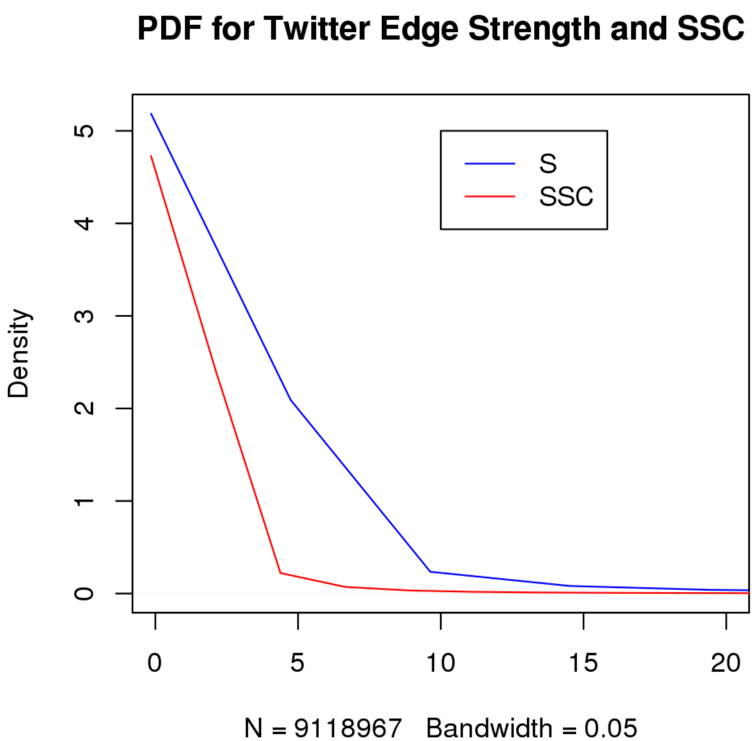
- SSC and total strength S seem to be (mostly) bounded by constant
- SSC seems to (mostly) be a good approximation to S

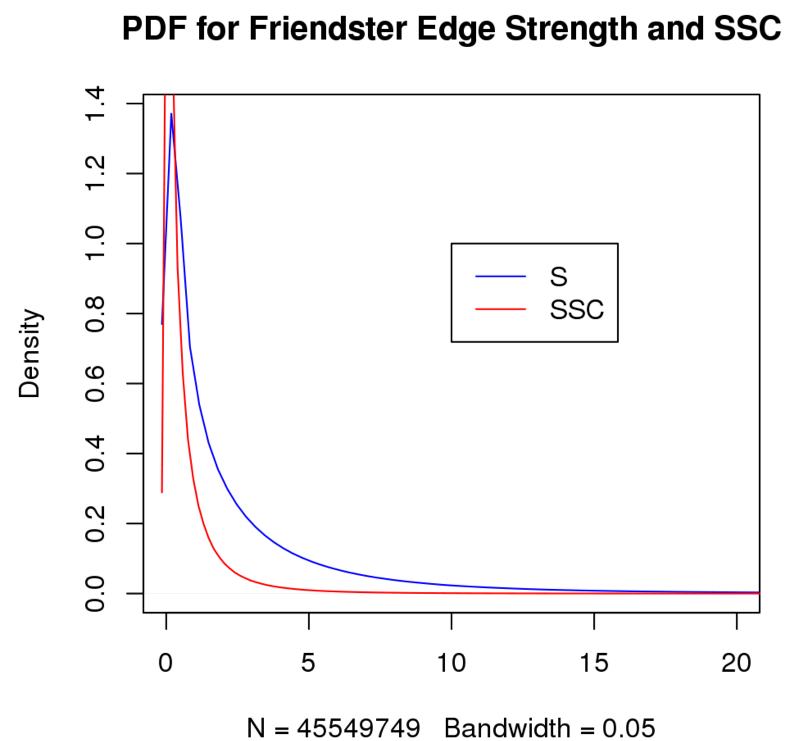




More Distributions

- Larger social networks





Cleaning Non-Human Nodes

- We assume $s \leq \sqrt{\frac{D}{d}}$ for all/most vertices
- Constant D will depend on the network
- Remove edges between nodes with s above this curve
- Selecting D
  - Compute average SSC average μ and standard deviation σ
  - D = μ + kσ for user-defined parameter k
- We use k=3

- We use only reciprocated edges

Why not remove whole vertex?

- Sometimes small number of vertices have a large fraction of edges
- Conservative

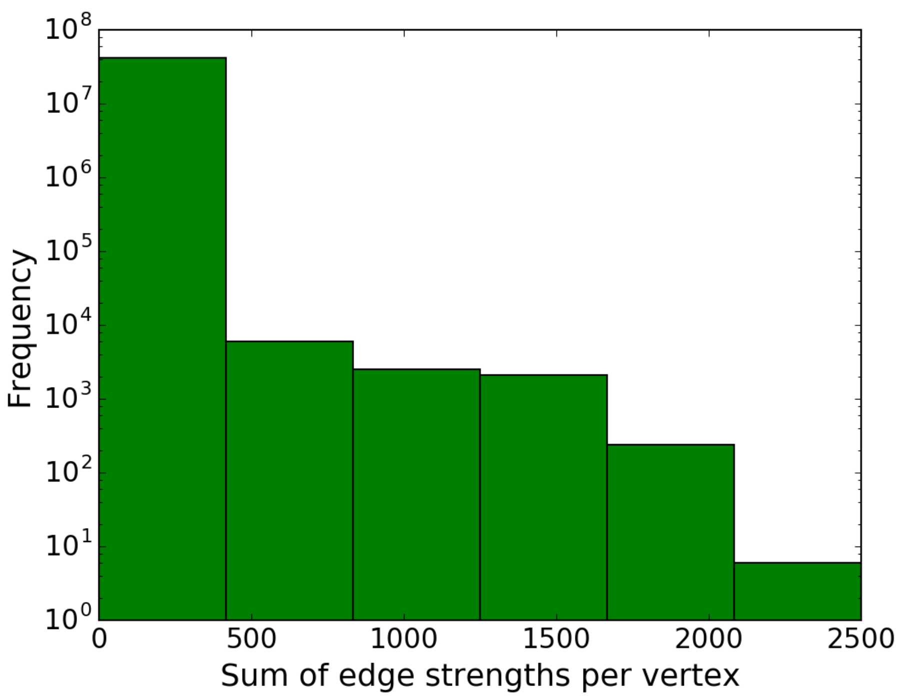
Network	percentage of vertices removed	percentage of edges removed
com-youtube($12\bar{\sigma}$)	0.01%	2.5%
com-youtube($6\bar{\sigma}$)	0.11%	10.76%
com-youtube($3\bar{\sigma}$)	1.18%	**32%**
ljournal-2008($12\bar{\sigma}$)	0.05%	1.57%
ljournal-2008($6\bar{\sigma}$)	0.14%	3.13%
ljournal-2008($3\bar{\sigma}$)	0.36%	5.38%
twitter-2010($12\bar{\sigma}$)	0.02%	**26.4%**
twitter-2010($6\bar{\sigma}$)	0.046%	**34.3%**
twitter-2010($3\bar{\sigma}$)	0.048%	**34.7%**

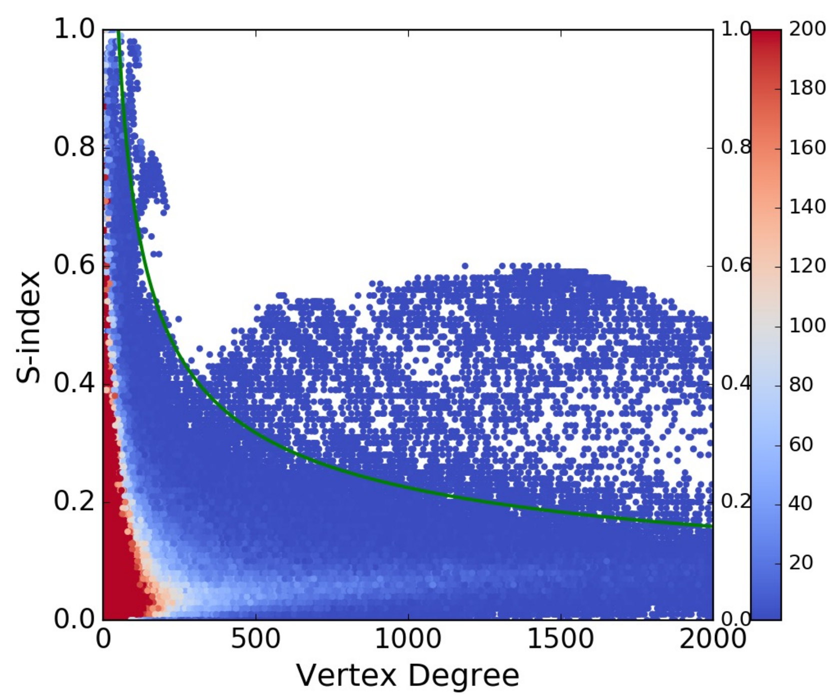
Why not remove whole vertex?

- Twitter is one social network where we can look up accounts
- Initial validation:
  - Strange connectivity:
    - A musician from a late-night show
    - A frisbee golf company (in New Jersey?)
    - Filmchair
    - Another unrelated Canadian company
    - Etc
- Conjecture: They paid a company to manage their Twitter accounts and the company connected them all

Twitter

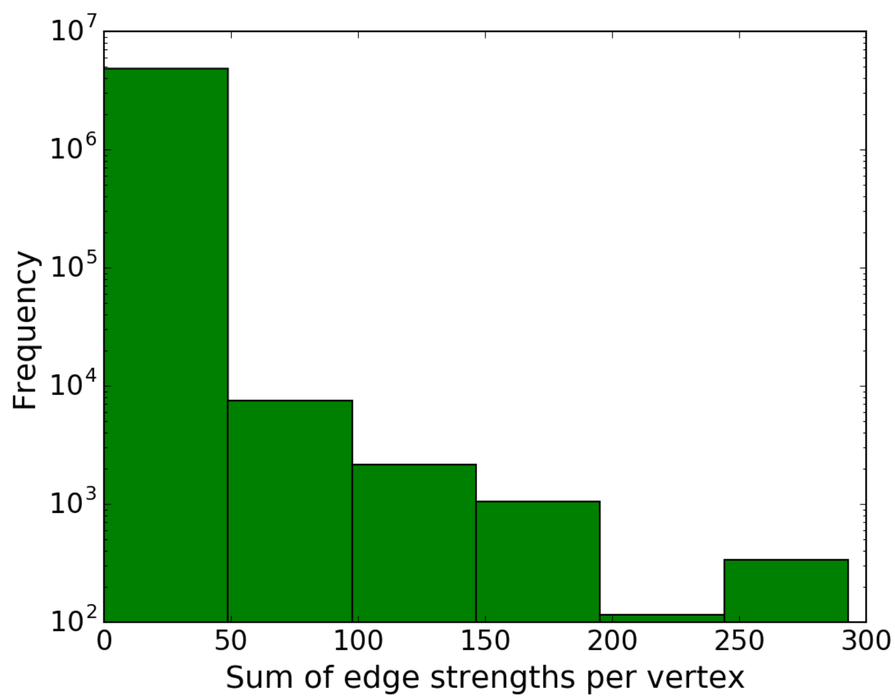
- 41.5M nodes, 266M reciprocated edges, D_G = 50

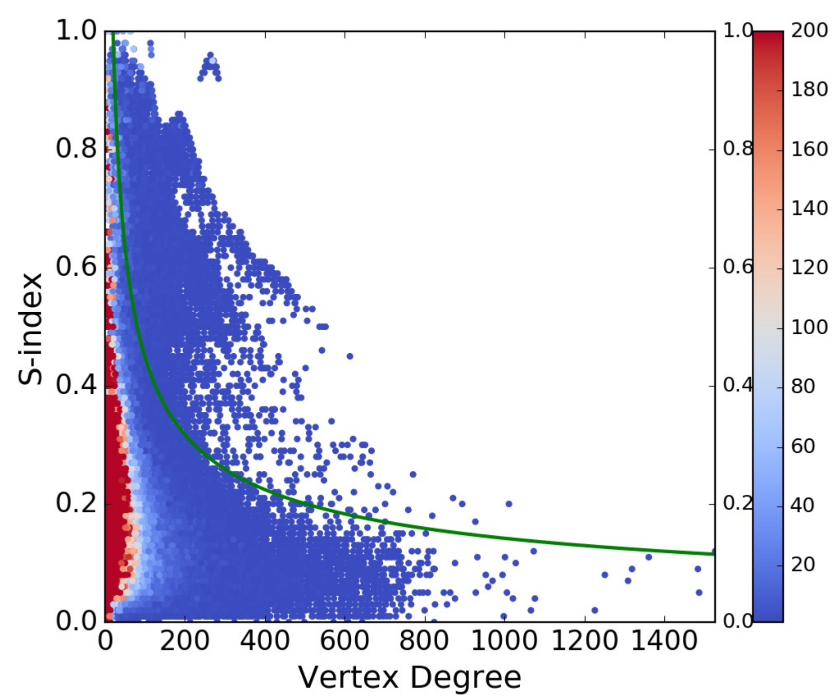




LiveJournal

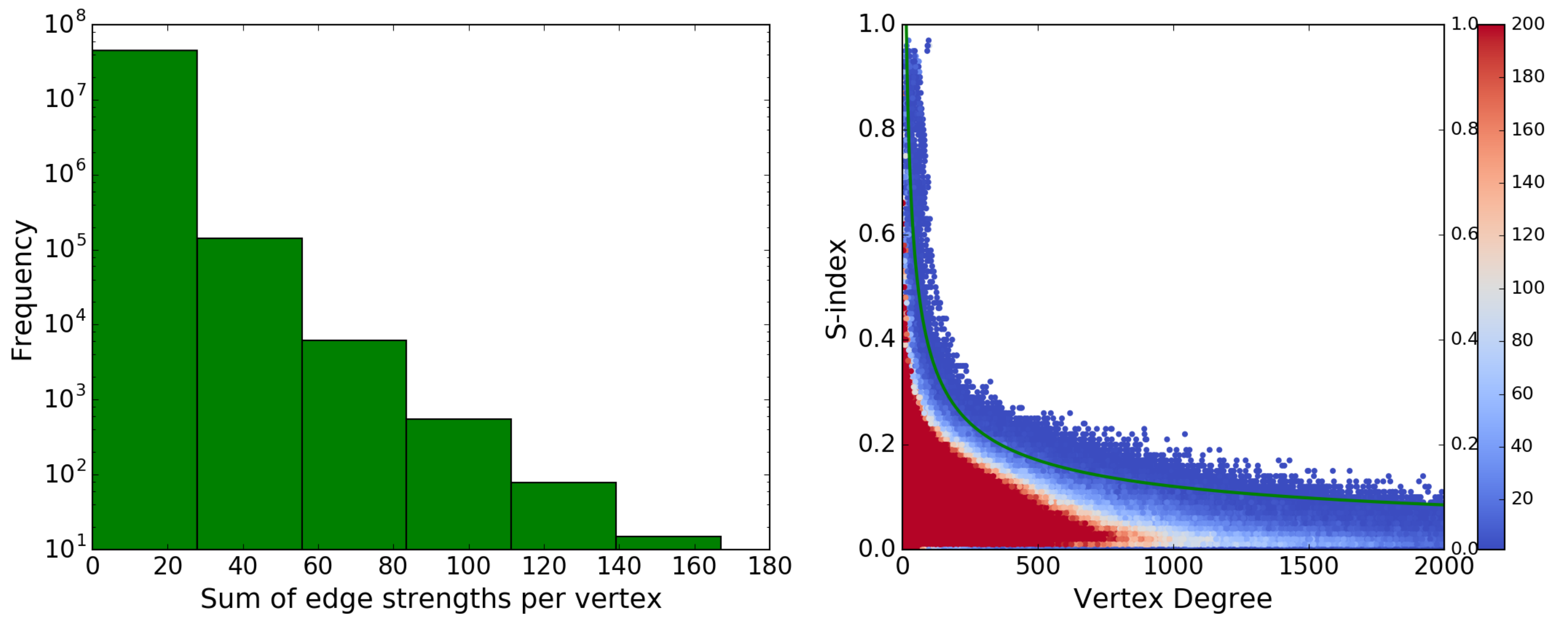
- 4.8M nodes, 25.6M reciprocated edges, D_G=20





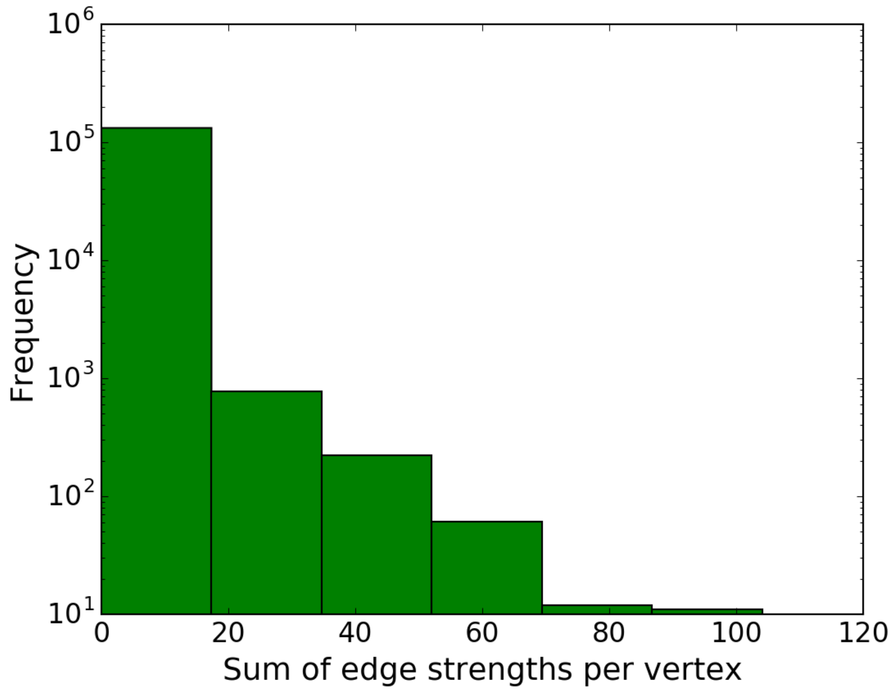
Friendster

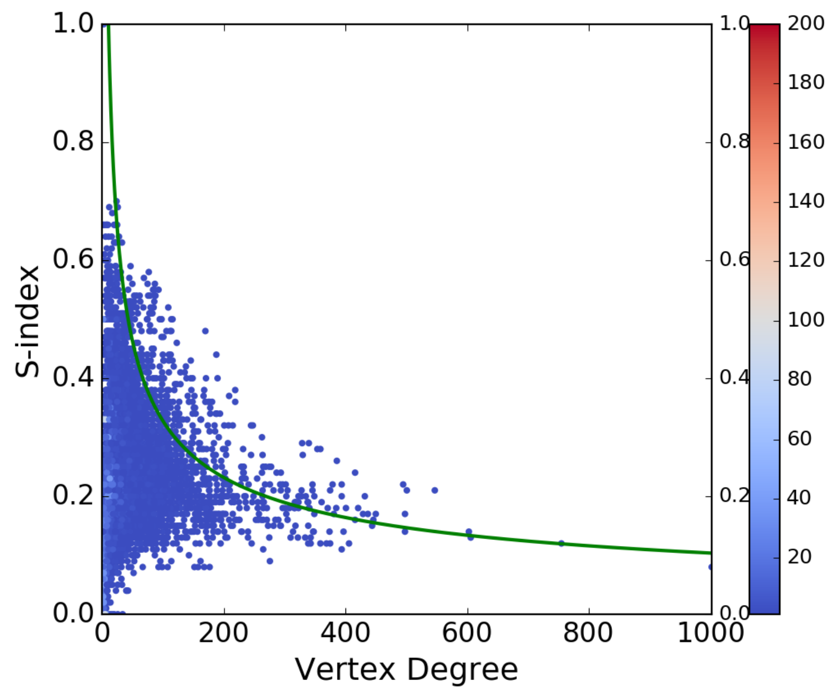
- 65.6M nodes, 1.8B reciprocated edges, D_G=14



Ca-AstroPh (citation)

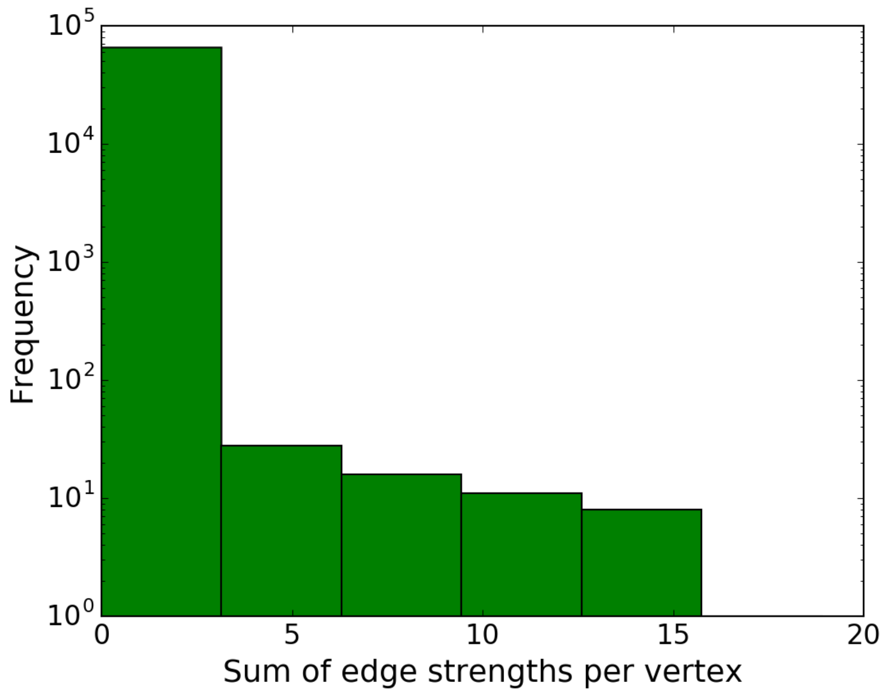
- 133K nodes, 198 reciprocated edges, D_G=11

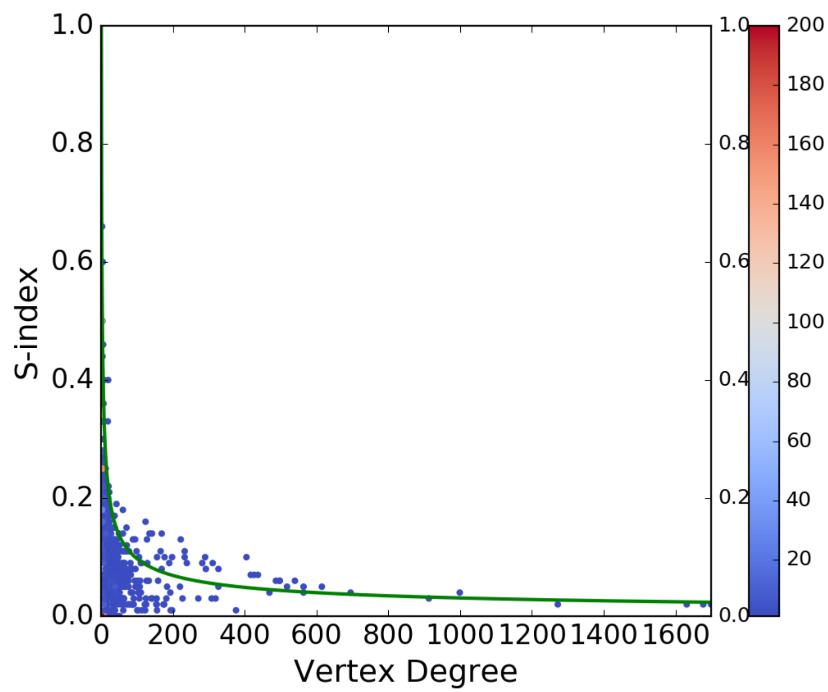




Caida (web)

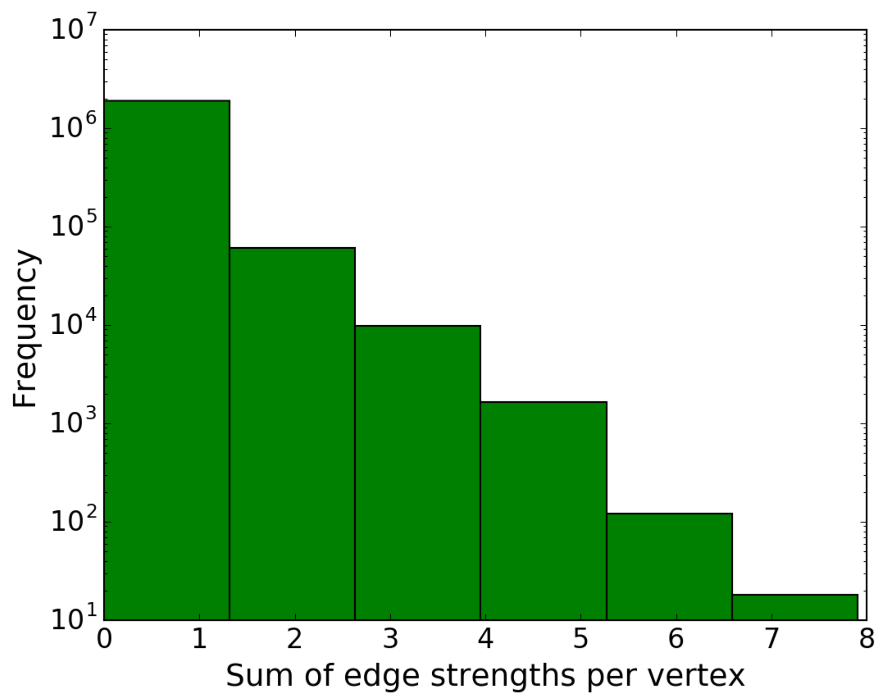
- 26K nodes, 53K reciprocated edges, DG=0.9

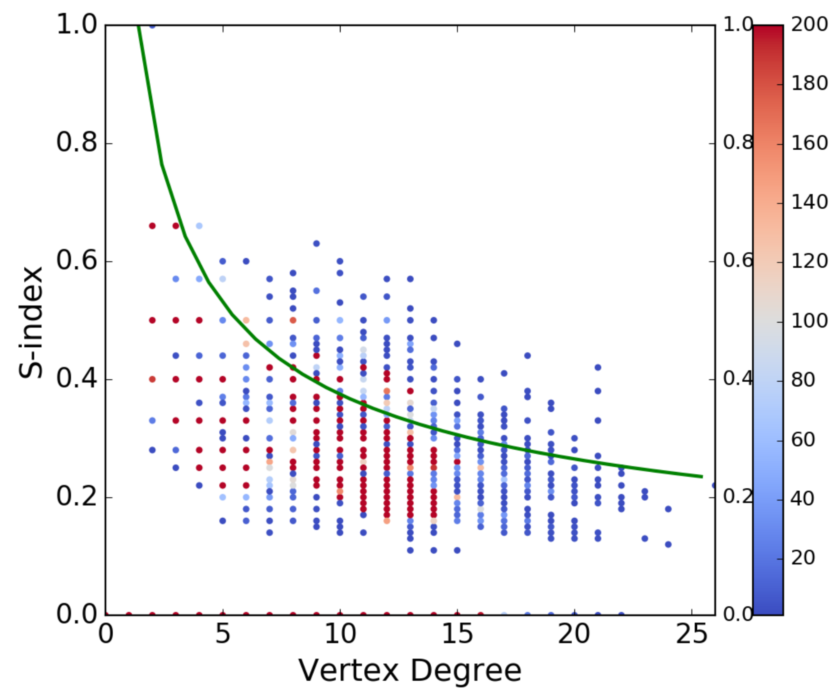




CA-RoadNet

- 2M nodes, 5.5M reciprocated edges, D_G=1.3



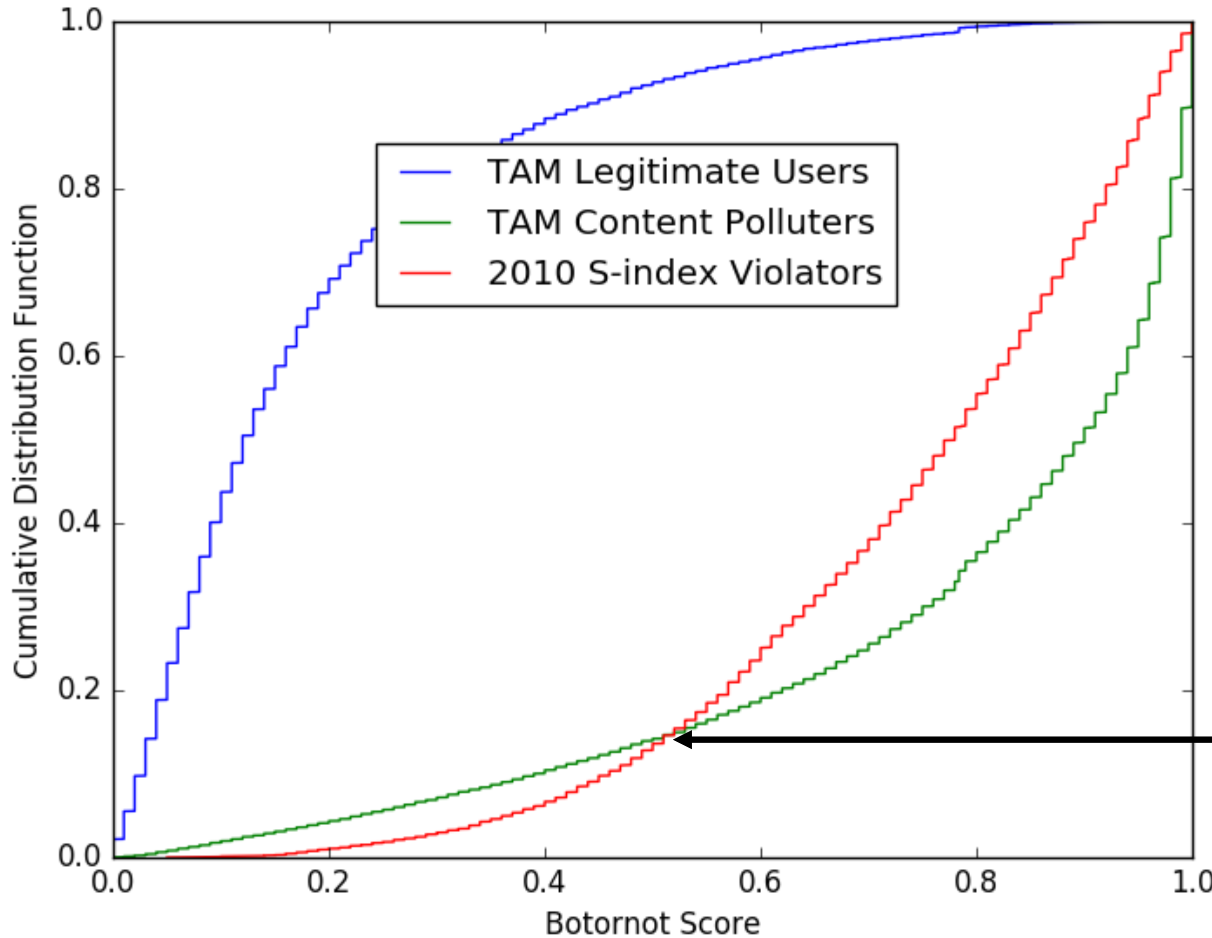


Validation 1

- Twitter has an API to look up users
  - O(10) account look ups/min
  - O(1) follower list look up/min
- First test "Bot-ness"
- Compare to
  - Texas A&M hand-labeled set of Twitter nodes (30K)
    - Human inspection
  - BotOrNot Scores (https://truthy.indiana.edu/botornot/)
    - BotOrNot uses account features
  - Our s-index violators came only from topology

Question: do our strength-index violators and the Texas A&M (TAM) Ground truth nodes have similar Botornot score distributions?

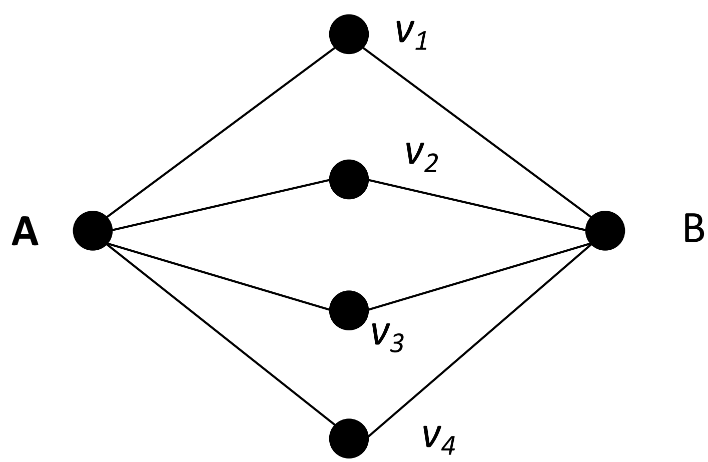
Bot-ness Results



~90% have BotOrNot Scores > 0.5

Validation 2: Order of Following

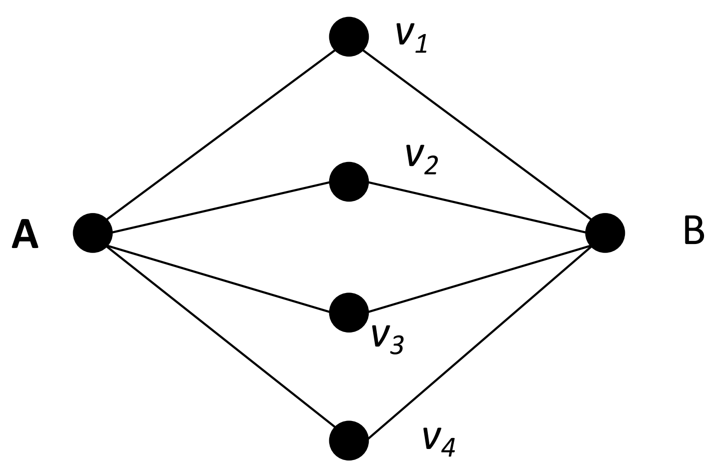
- An automated system might add followers in a given order
  - Adding whole botnet
  - Adding a new paying customer (add them to end of list)
- Consider order of adding shared neighbors



A's order v_1, v_2, v_3, v_4
B's order v_2, v_3, v_1, v_4

Validation 2: Order of Following

- Consider order of adding shared neighbors
- Longest common subsequence of 2 length-n sequences
  - If added intentionally in order (automated) expect $\Theta(n)$
  - Random is $2\sqrt{n}$
    - Expect human to be more random



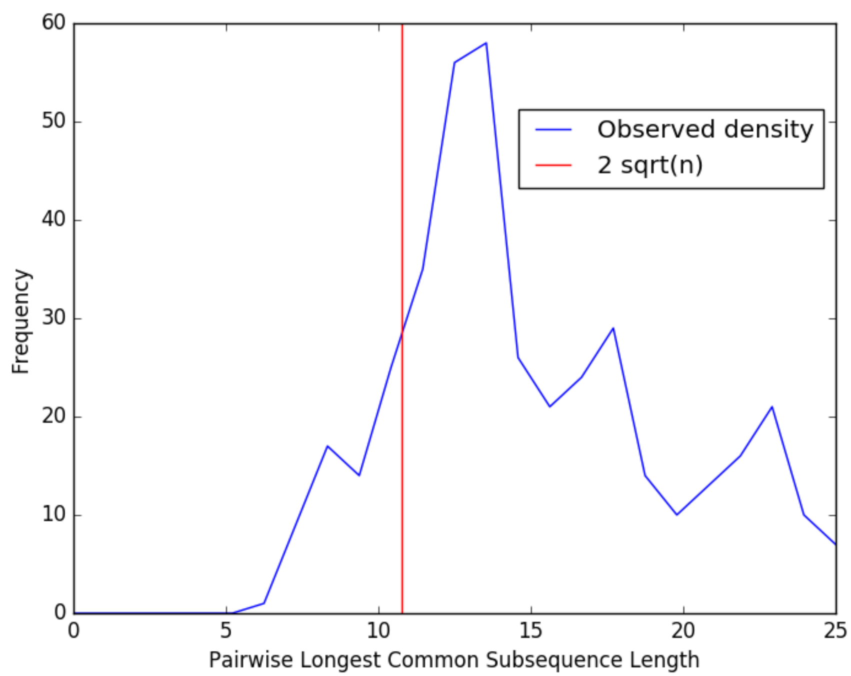
A's order v_1, v_2, v_3, v_4

B's order v_2, v_3, v_1, v_4

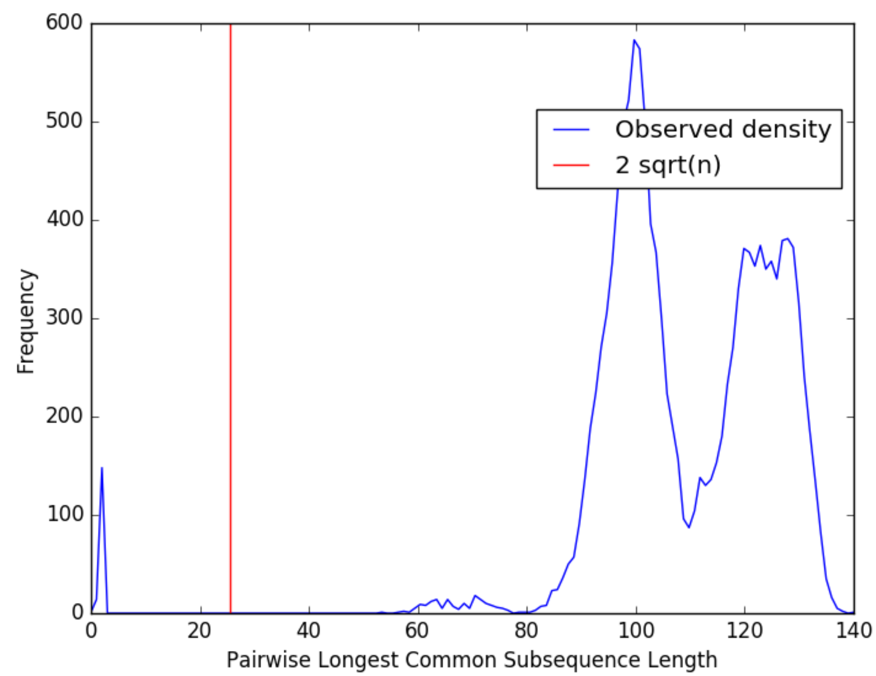
LCS is v_2, v_3, v_4

Order-of-Following Results

- Violators: largest clique 318 (in 2010), now 164
- Largest clique in non-violators was a small weather bot network
- Second-largest clique in non-violators 53 (in 2010), now 29



Non-violators

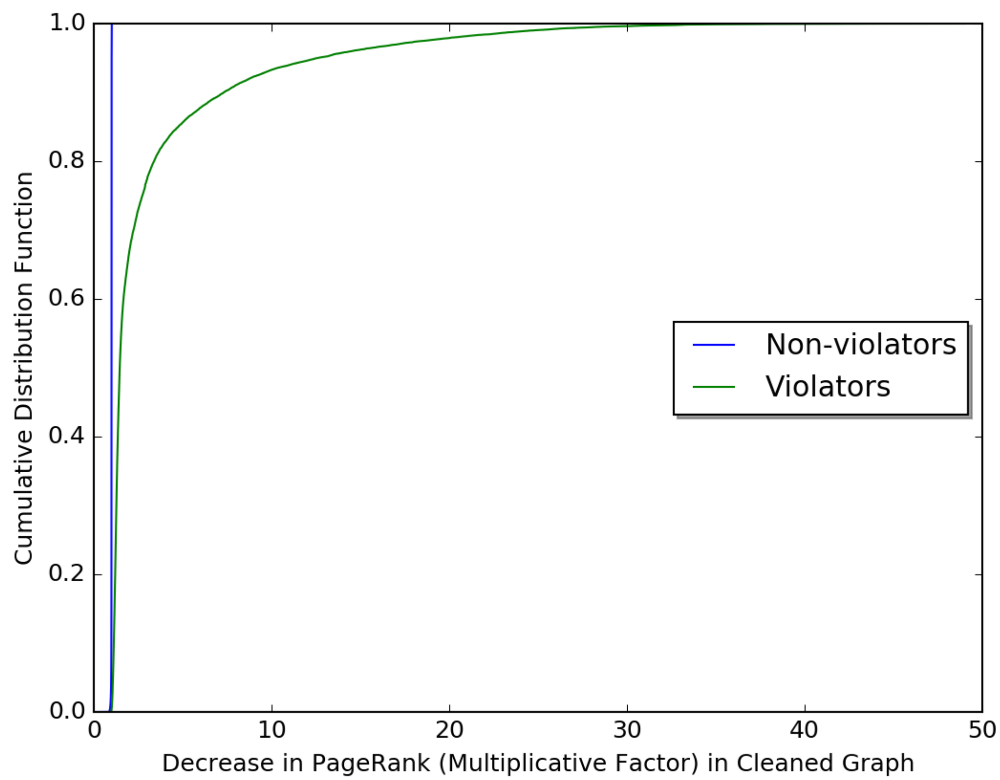


Violators

Consequences: PageRank

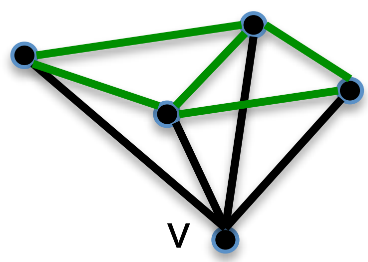
- People with access to real content on more social networks will need to further validate

- Does the cleaning matter?
- Yes for an algorithm like PageRank
  - 45% of violators have 2x decrease in cleaned graph
  - 16% decrease 5x



Clustering Coefficients

Fraction of wedges that close to a triangle



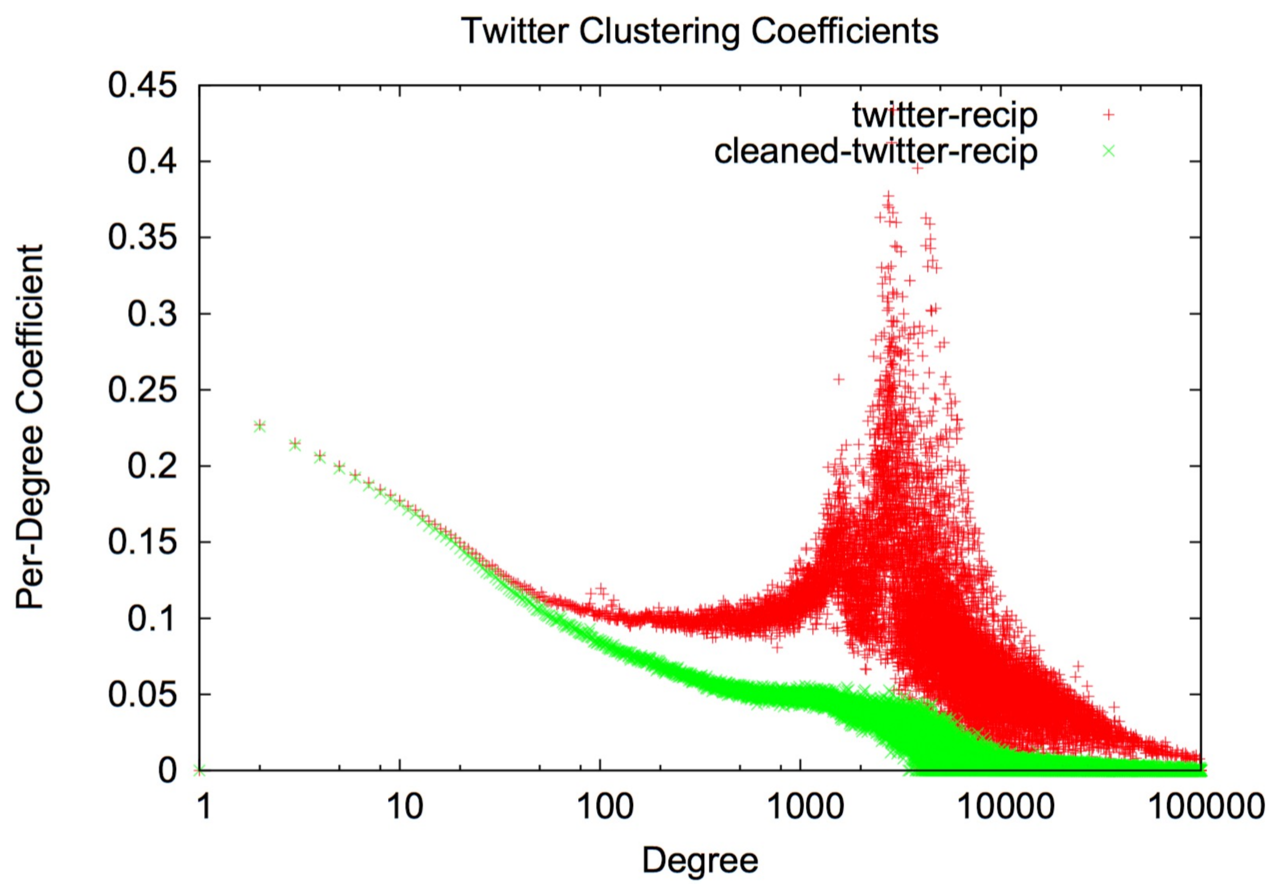
Clustering coefficient (CC) of v =
Fraction of related neighbors.

$$\frac{\text{\# triangles on v}}{\text{\# wedges on v}} = \frac{5}{6}$$

$c_{avg}(d)$ = Average CC over nodes of degree d.
Global CC = average over all nodes v

Consequences: Clustering Coefficients

- Clustering coefficients are a structural property
- The graph generator BTER uses only degree distribution and per-degree clustering coefficients



CHANTS Website

- CHANTS = Cleaner Human-Amplified Network Test Set
- Code to running cleaning on your own social network data set
- "Cleaned" versions of public data sets:
  - Twitter 2010
  - YouTube
  - LiveJournal
  - Pokec
  - Friendster
  - Orkut

https://www.cs.unm.edu/~socnet/CHANTS.html

Social-Networks Summary

- A possible tool for cleaning some non-human behavior from some social networks.
  - conservative
- Social network structure enables more efficient algorithms in theory and practice, but requires human-only networks.
- This seems to be different from bot detection methods
  - Bad edges on non-bot nodes
- We won't be able to validate the other networks
- Theory implications are wide open

J. Berry, C. A. Phillips, and J. Saia, "Making Social Networks More Human: a Topological Approach," Statistical Analysis and Data Mining The ASA Data Science Journal, Vol. 12, No. 6, pp. 449-464, December, 2019.

Story 2: Security Challenge

- Systems sacrifice security for I/O efficiency
  - Example: Microsoft Word “fast save” appends edit log
  - Adversaries can recover old versions of documents

- Hide the history of a data structure on disk
  - Order of arrival
  - No idea if there has ever been a deletion

History-Independent Data Structures

- An added level of protection for data on disk
- An adversary who acquires the disk and examines memory cannot determine anything more than API would give
- If the adversary can examine the disk cannot determine:
  - Order elements arrived
  - If any data has been deleted
- Order information can reveal sources, policy, etc.
- One potential motivation: drones

From: Wikipedia

History Independence (HI)

- Strong history independence gives guarantees if the adversary sees the data representation multiple times
  - Requires a canonical representation
  - Expensive
    - Provably cannot achieve amortized o(N) operation cost whp
- Weak history independence protects against a one-time theft
  - Representation is drawn uniformly at random from a given large structured set
  - Can be much more efficient
  - The right model if a disk can only be stolen once

Hint at some details

- Oblivious adversary for analysis: sets order of operations, but does not know the random tosses of the data structure

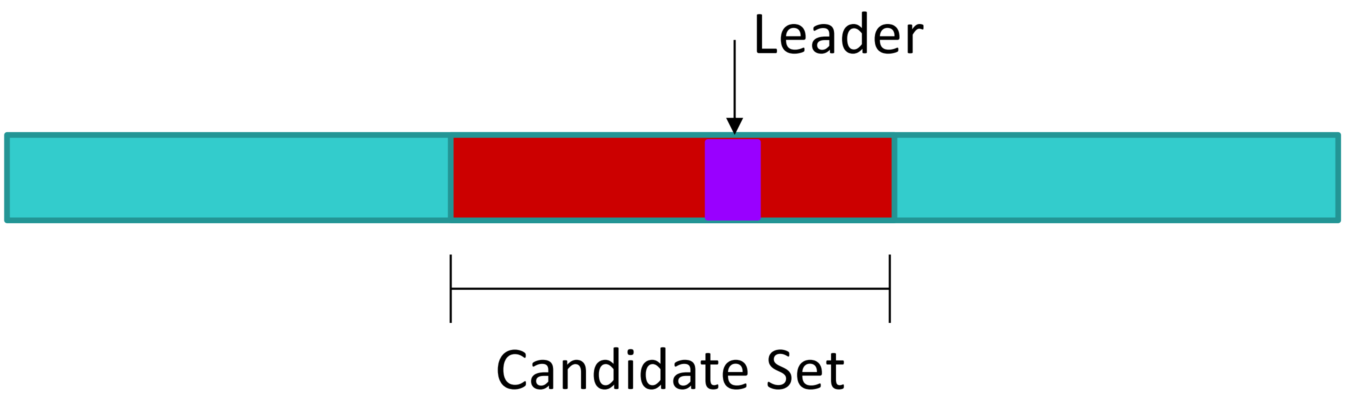
- Search tree
- All the elements are in the leaves (many per leaf) on disk
- Randomization involves how many elements in each leaf

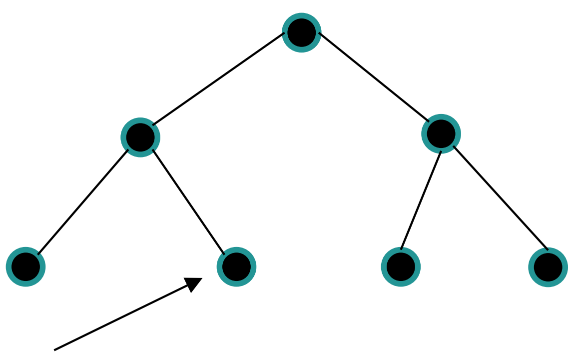
To start, size/storage allocation

- For N elements, allocate array size |A| from N to 2N-1 uniformly
- For any insert/delete reallocate with probability $\Theta\left(\frac{1}{|A|}\right)$

Key ideas

Recursive stick breaking





Leaves have log N elements
Always packed left

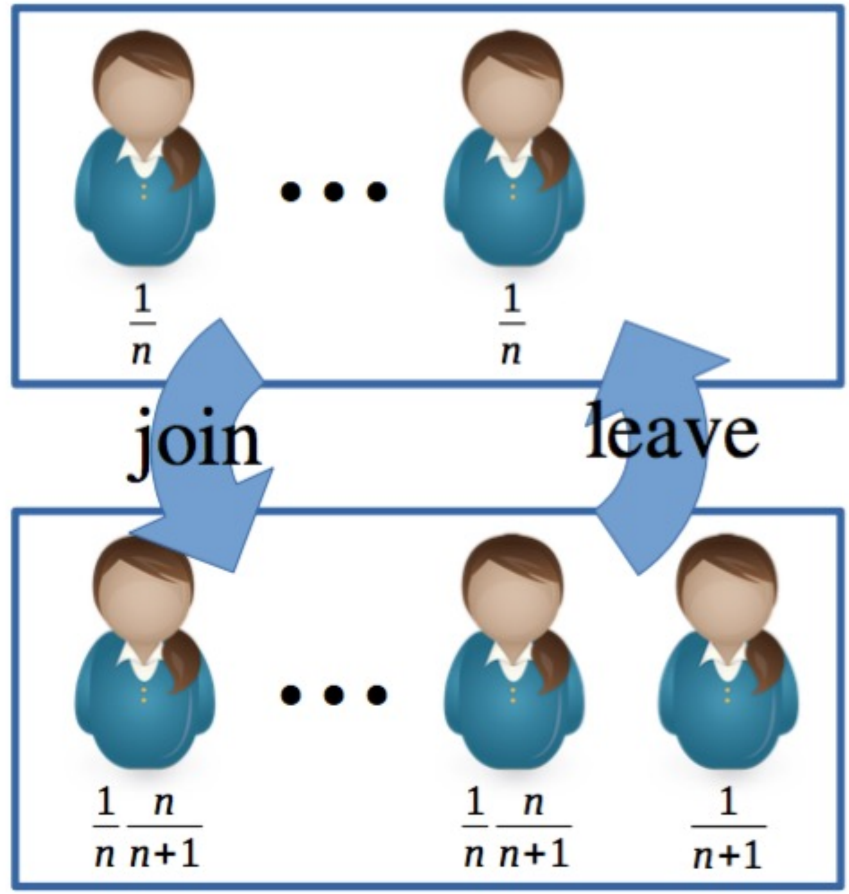
At any point, if the adversary looks at the disk, the layout distribution corresponds to the distribution from this full rebuild process.

Reservoir Sampling with Joins and Leaves [Vitter '85]

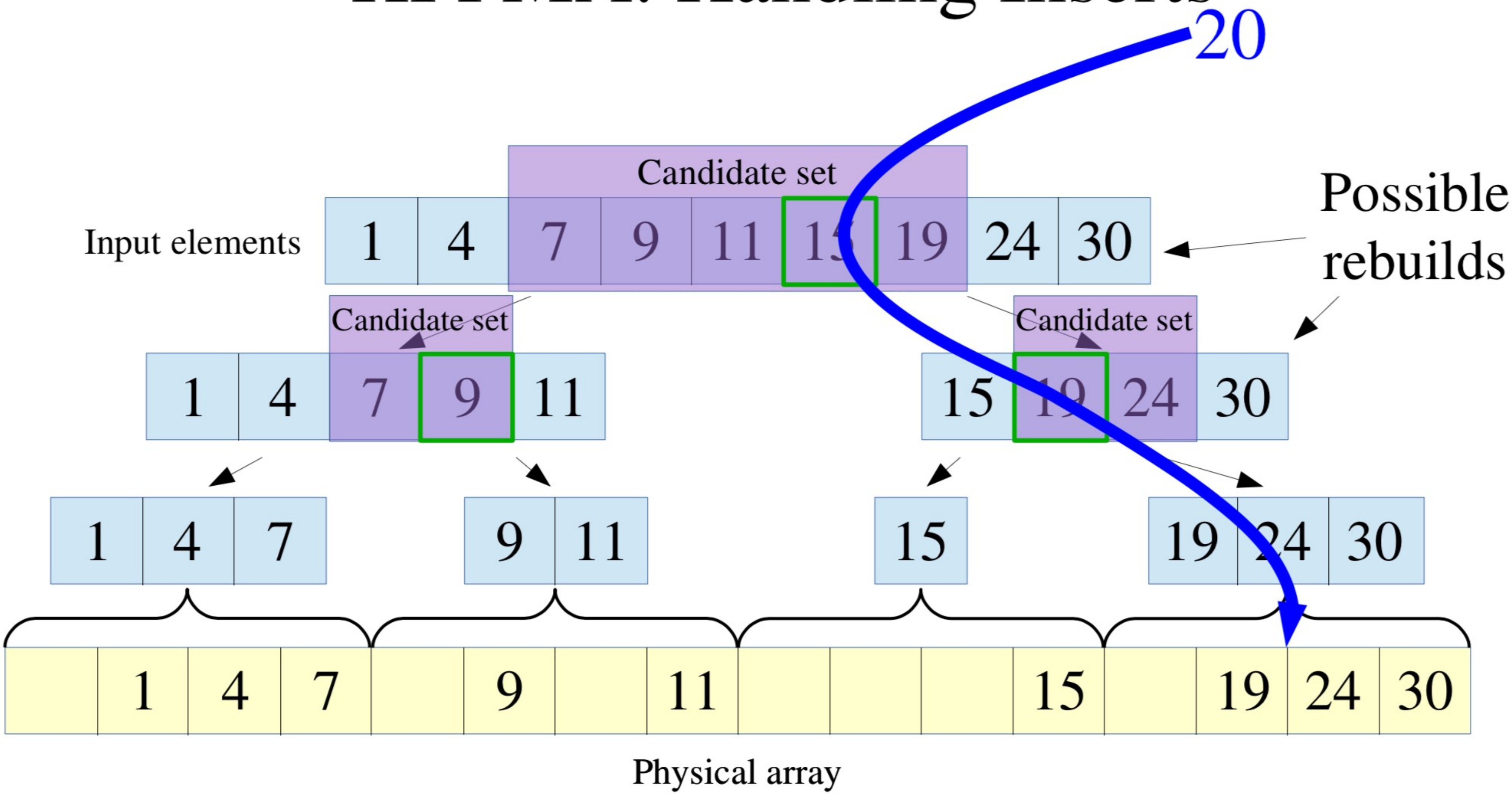
- Two goals:
  - Maintain a club leader uniformly randomly from all current club members
  - Make leader changes rare as members join and leave

1. Elect new member w/ prob $1/(n+1)$
2. Elect new leader when leader leaves

Prob[leader changes] $\approx 1/n$



HI PMA: Handling Inserts

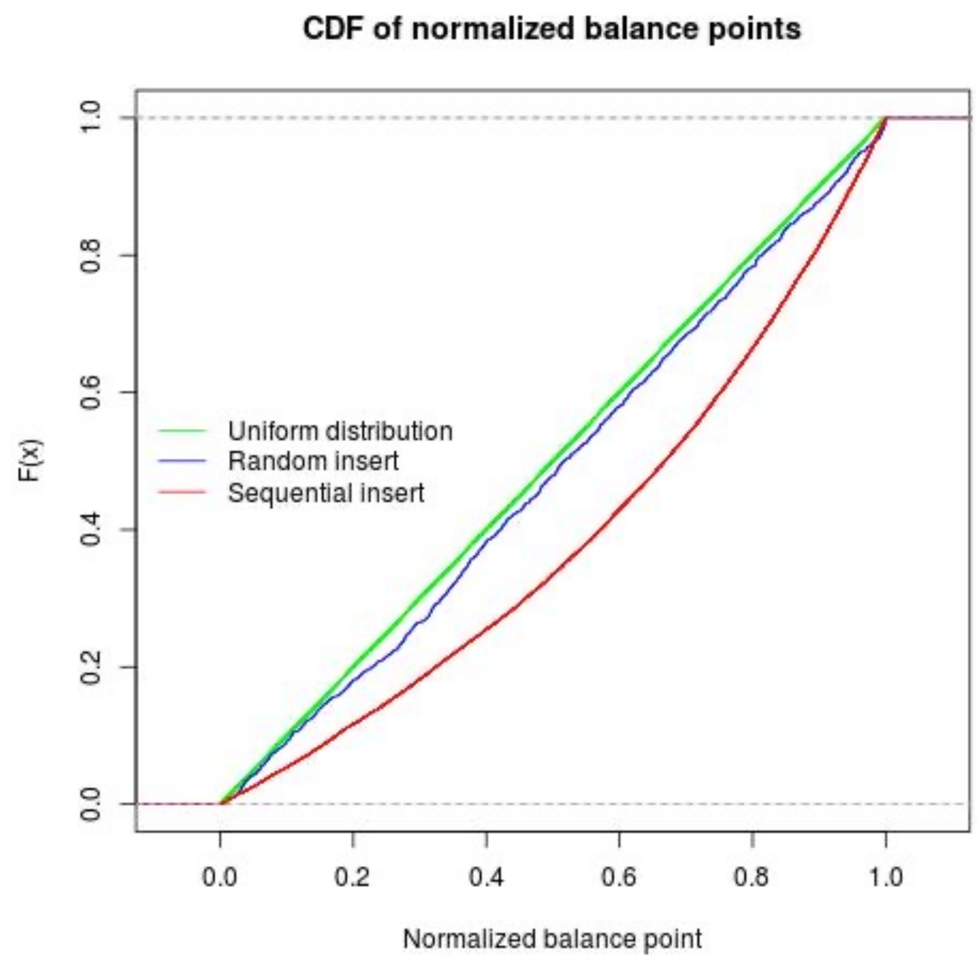


Validation

- HI data structures cost something (but only a constant factor in theory and about 7x on initial experiments)
- If there is any error in the implementation, could lose HI property
  - History independence is delicate
- How to validate an implementation?
- Easy to check the correctness of dictionary aspects
  - Verify set of keys is correct over many insertion/deletion tests
- How to test that the bit representation always drawn from the distribution of representations immediately following a clean rebuild?

First Simple Test

- Balance value at the root
  - Just compare for equal allocation sizes
  - Rank of balance point within candidate set should be uniform
  - Insert keys 1 to 100,000 from empty start (in some order)



The Fix

Fairly small errors in logic or coding can destroy the history independence of the implementation even though dictionary performance is correct.

```
# Old logic:
Insert(e)
...
  If e  in Candidate set
      If (lottery for e)
            e  is new balance element and return

   If current balance knocked out
       Pick new balance & return.

# New working logic
Insert(e)
 ...
 If current balance knocked out
       Pick new balance & return.

 If e in Candidate set
      If (lottery for e)
            e is new balance & return
```

More General Validation

PMA Layout distributions from 3 procedures:
X distribution: Build a PMA on set S from scratch.

Y distribution:
1) Pick an arbitrary element $y \in S$,
2) Build an HIPMA from scratch on $S - \{y\}$,
3) Insert y into the PMA.

Z distribution:
1) Pick an arbitrary element z not in S
2) 2) build an HI PMA from scratch on $S \cup \{z\}$
3) 3) delete z from the HI PMA.

If insertion and deletion are implemented correctly, then all three distributions X , Y , and Z should be identical.

More General Validation

X = Build S. Y = insertion to S. Z = deletion to S

If insertion and deletion are implemented correctly, then all three distributions *X* , *Y* , and *Z* should be identical.

Kullback-Leibler Divergence (like testing for a fair die)

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

- Smallest size non-trivial data structure
- Trials in parallel
- Can use computed probabilities for X

History-Independence Summary

- Can have weak history independence at no asymptotic cost

Open research questions:

- Are there clever statistically rigorous tests that are tractable?
- Other Applications?

Michael A. Bender, Jonathan W. Berry, Rob Johnson, Thomas M. Kroeger, Samuel McCauley, Cynthia A. Phillips, Bertrand Simon, Shikha Singh, and David Zage. Anti- persistence on persistent storage: History-independent sparse tables and dictionaries. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS ’16, pages 289–302, New York, NY, USA, 2016. ACM.

Story 3: Constrained Randomized Rounding

- k-of-N selection: Given N variables with $0 \leq x_i \leq 1$ and

$$\sum_i x_i = k$$

- Select k of the x_i such that probability of selecting variable i is reasonably related to x_i.
- Motivation: linear programming relaxation of integer program
- In multiple applications, this selection is main (only) decision
  - Sensor placement (e.g. in water networks)
  - Mobile sink scheduling for wireless networks
  - Enforcing node degree in graph generation

Rounding with One Cardinality Constraint

- Doerr (2004), motivated by Srinivasen (2001)
- Finds a randomized rounding y_i such that:

$$\Pr(y_i = 1) = x_i$$

$$\sum_i y_i = k$$ (Respects cardinality constraint)

Simple (base) Case

- All $x_i \in \{0, 1/2, 1\}$
- Let *X* be the set of x_i with value ½.

- |X| is even because $\sum_i x_i = k$ and *k* is integer.

- Pair elements of *X*: (x_i, x_j)
- Set (y_i, y_j) = (1,0) or (0,1), each with probability ½.

$$(0,1) \xleftarrow{\frac{1}{2}} (\tfrac{1}{2}, \tfrac{1}{2}) \xrightarrow{\frac{1}{2}} (1,0)$$

General Case

- Do base case for lowest-order bit r (most to right of binary point)

$$(x_i - 2^{-r}, x_j + 2^{-r}) \xleftarrow{\frac{1}{2}} (x_i, x_j) \xrightarrow{\frac{1}{2}} (x_i + 2^{-r}, x_j - 2^{-r})$$

- After this operation, rightmost bit in place r-1
- Iterate to compute y in $O(Nr)$ time, where N is the number of x_i and r is the initial rightmost bit.

- Numerical issue: In (floating point) practice, $\sum_i x_i \approx k$ not integer.
- Even going to 1000 bits doesn't necessarily fix this (and slow)
- Open: Get this to work in practice. Can we efficiently convert to binary representation?

Thank you to my Collaborators

- Jon Berry (Sandia National Laboratories)
- Michael Bender (Stony Brook University)
- Rob Johnson (VMWare Research)
- Justin Jacobs (Sandia National Laboratories)
- Thomas Kroeger (Sandia National Laboratories)
- Samuel McCauley (Williams College)
- Jared Saia (University of New Mexico)
- Bertrand Simon (Universität Bremen)
- Shikha Singh (William College)
- David Zage (Intel)

Final Comments

- Not all beautiful, elegant, simple algorithms work well, as is, in practice.
- Find clever examples and methods to test the correctness of an implementation before testing performance (even though you really, really want to test performance because that deadline is looming).
- Try out the CHANTS data set.
- Special properties of national-security applications also lead to interesting theory problems (new structure/constraints, with motivation).

Back up Slides

History-Independent Data Structures [Naor & Teague '01]

[Blelloch & Golovin '07] [Buchbinder & Petrank '03] [Bajaj, Chakrabati, Sion '15] [Bajaj & Sion '13] [Molnar, Kohno, Sastry, Wagner '06] [Moran, Naor, Segev '07] [Naor, Segev, Wieder '08] [Roche, Aviv, Choi '15] [Tzouramanis '12] [Golovin '08, '09, '10]

- Bit representation reveals no additional info about past states of the data structure
- Example:

{A, B, C}

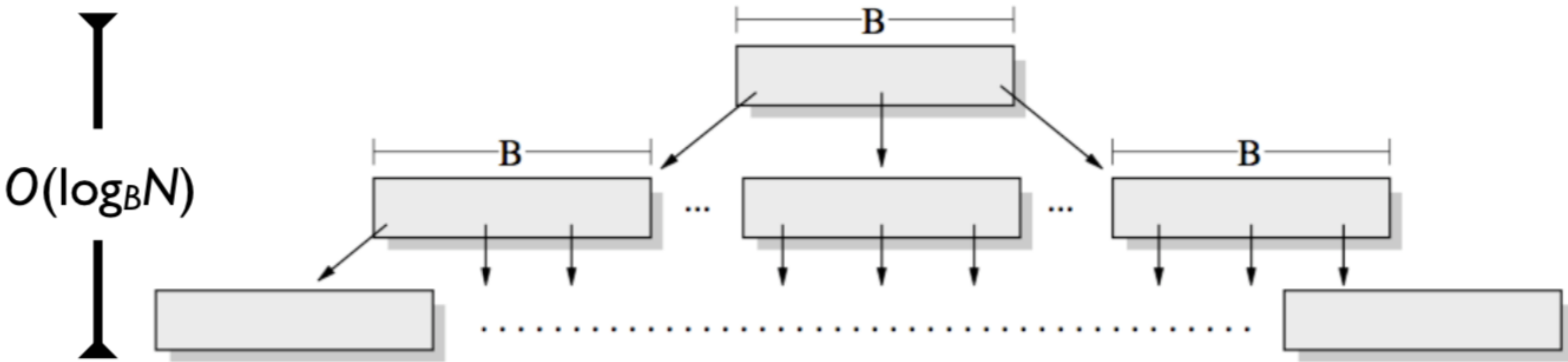
Observer cannot infer sequence of operations leading to current state

1.Insert A	1.Insert C
2.Insert B	2.Insert B
3.Insert C	3.Insert A
4.Insert D	
5.Delete D	

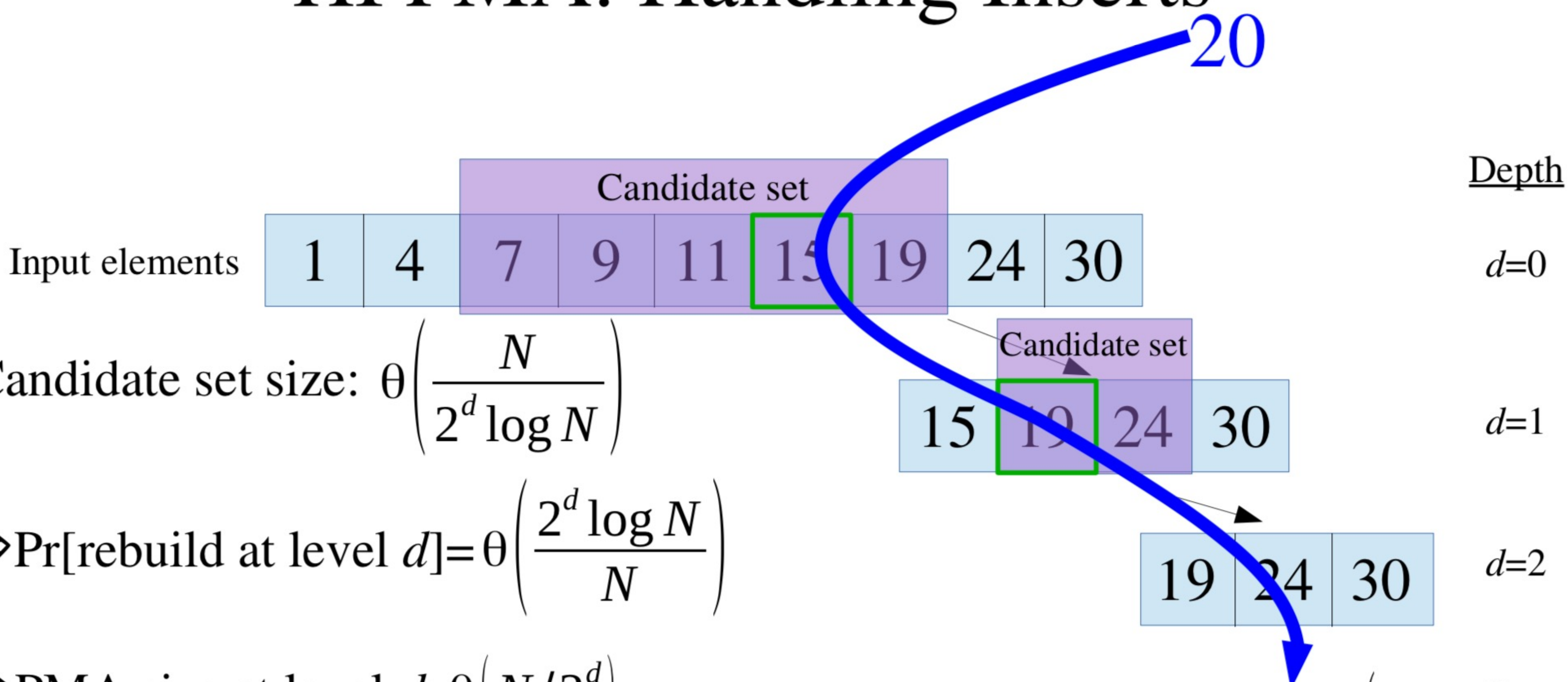
History-Independent Dictionaries

- Skip lists. External memory block size B, n items
  - Insert, delete, search: $O(\log_B n)$
  - Range search with k items in range: $O(\log_B n + k/B)$ block
  - Amortized, with high probability: $1 - O\left(\frac{1}{n^c}\right)$
  - Optimal
- Previous work for HI skip lists: insert $\Theta(\log n)$
- Cache-oblivious B-trees
  - Same bounds except inserts are (optimal) $O(\frac{log^2 n}{B} + \log_B n)$
  - $O(n)$ space
  - Experiments show small slowdown

B-trees



HI PMA: Handling Inserts



Candidate set size: $\theta\left(\frac{N}{2^d \log N}\right)$

$\Rightarrow$Pr[rebuild at level d]$=\theta\left(\frac{2^d \log N}{N}\right)$

$\Rightarrow$PMA size at level d: $\theta\left(N/2^d\right)$

$\Rightarrow$E[work for rebuild at level d]$=\theta\left(\log N\right)$

E[insert cost]$=\theta\left(\log^2 N\right)$