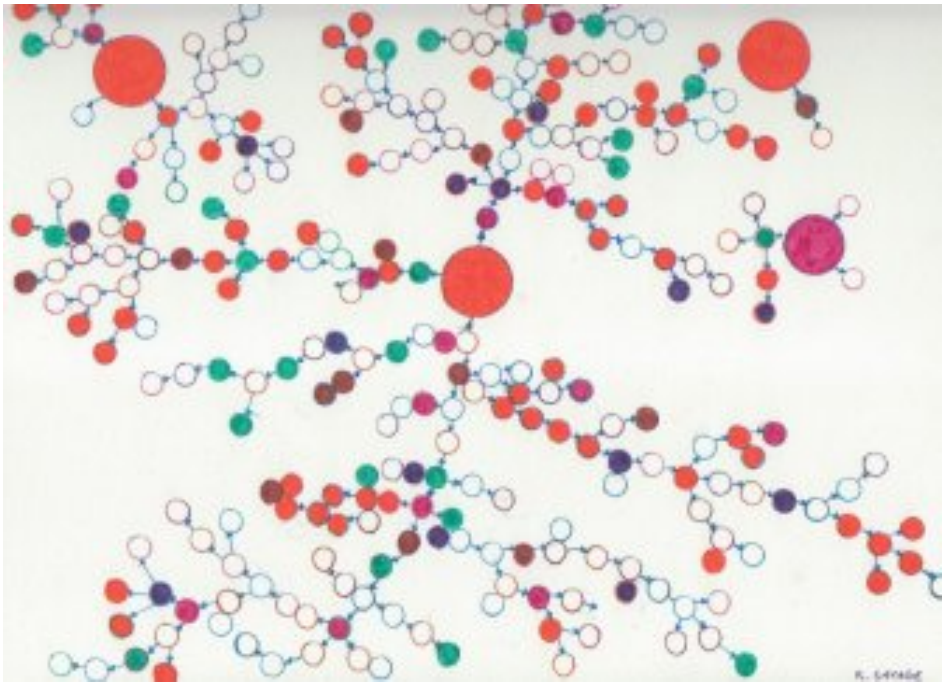
The Structure of Online Diffusion Networks

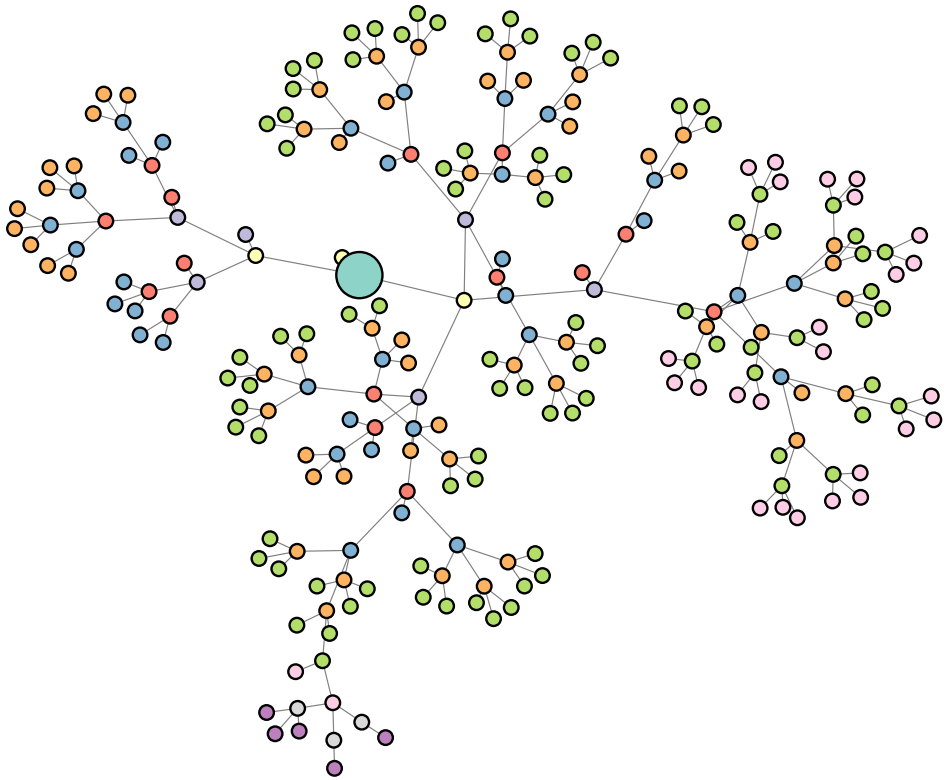
(or Why This Talk Won't Go Viral)



Joint work with Duncan Watts and Dan Goldstein
Microsoft Research – NYC

Information Diffusion:
How do ideas and products spread through cultures?

RDS



Bastos, F.I., 2009

Traditional Social Science Approaches

- Self-Reported Data: Surveys, Polls
- Aggregate Adoption Data
- Laboratory Experiments
- "Data-Free" Methods: Simulations, Theoretical Models, Rhetoric

A Large-Scale Empirical Approach

Observe a lot (millions/billions) of peer-to-peer transmissions of distinct products.

This used to be hard.

Information Diffusion:
How do ideas and products spread through cultures?

(Mostly) Direct – via URL tracking

- Yahoo! Kindness
- The Secretary Game
- Zync - A video sharing application

Indirect – via time-stamped adoptions over a known network

- YouTube Videos on Twitter
- News Stories on Twitter
- Friendsense – A Facebook Application
- Yahoo! Voice – A PC-to-Phone service

How do you analyze 1M+ diffusion events over a 1B+ edge network?

Answer: MapReduce

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

MapReduce

The Paradigm: Split-Apply-Combine

The Result: Programs scale transparently — size doesn't matter

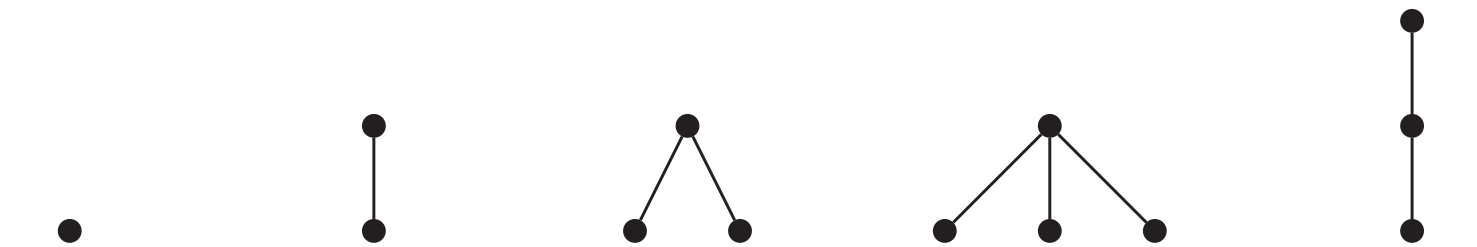
The conceptual simplicity of MapReduce masks the hard engineering (e.g., need to abstract away fault tolerance, synchronization, etc.).

Computing the Structure of Diffusion

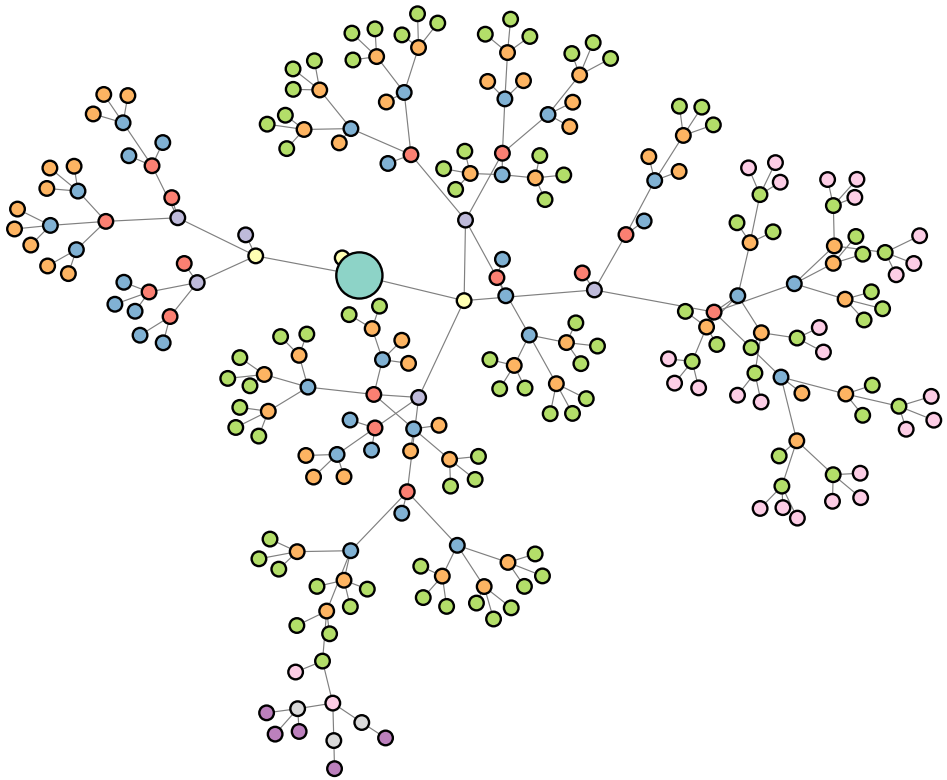
```
-- load and filter email messages
msgs = LOAD '$mail_msgs'
msgs = FILTER msgs BY (num_recips <= 50) AND (ispam == 0);
msgs = FOREACH msgs GENERATE date, sender, mailfuns.recips(to, cc, bcc) AS recips;

-- generate network edges
sent = FOREACH msgs GENERATE date, sender AS u1, FLATTEN(recips) AS u2;
sent = FILTER edges BY (u1 != u2);
received = FOREACH sent GENERATE date, u2 AS u1, u1 AS u2;

-- restrict to reciprocated edges
edges = COGROUP sent BY (u1, u2) INNER, received BY (u1, u2) INNER;
edges = FOREACH edges GENERATE
  group.u1 AS u1,
  group.u2 AS u2,
  COUNT(sent) AS sent,
  COUNT(received) AS received;

-- store results
STORE edges INTO '$output';
```

10 lines of code, 10 MapReduce Rounds, 1,000 compute nodes, and 1 hour later, we have the structure of diffusion on twitter.

The Structure of Diffusion on Twitter

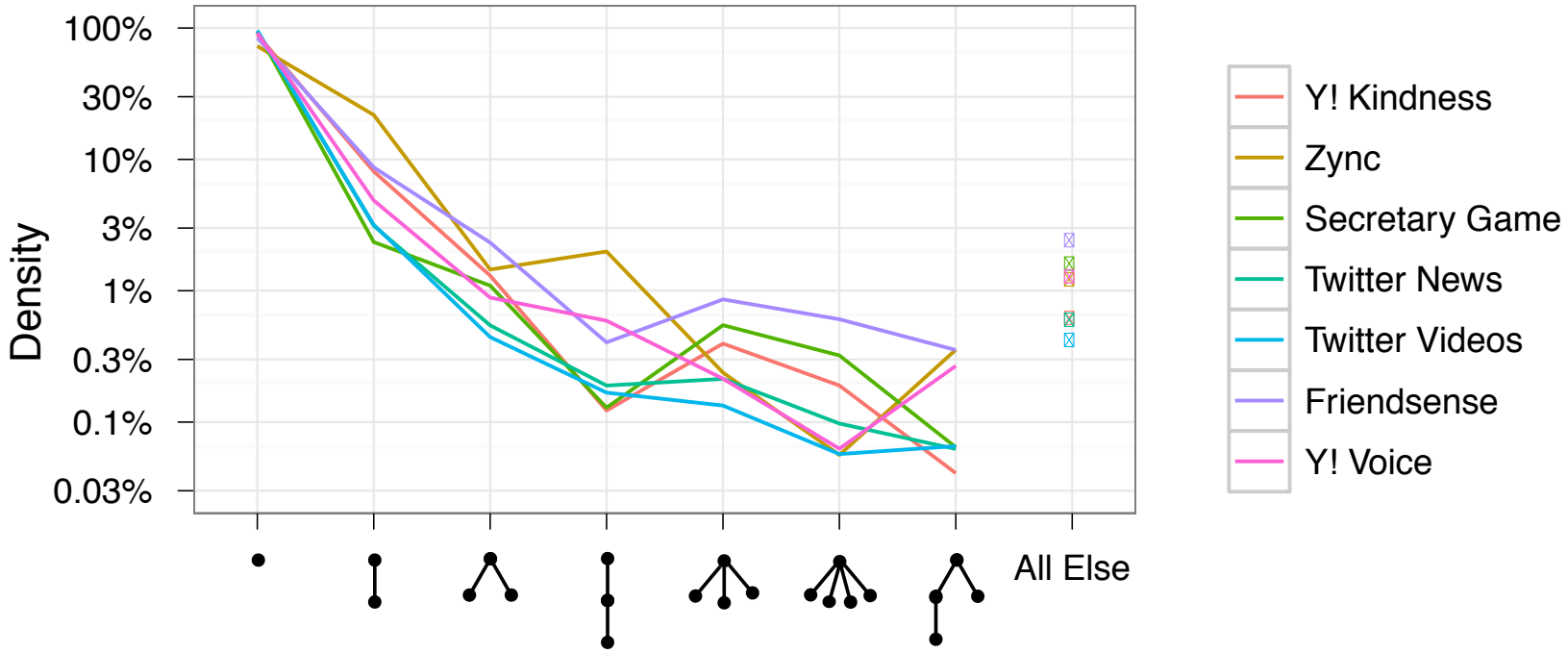


93% 5% 0.9% 0.3% 0.3%



Bastos, F.I., 2009

The Structure of Diffusion on Across All Seven Domains



73% - 95% of trees have no children

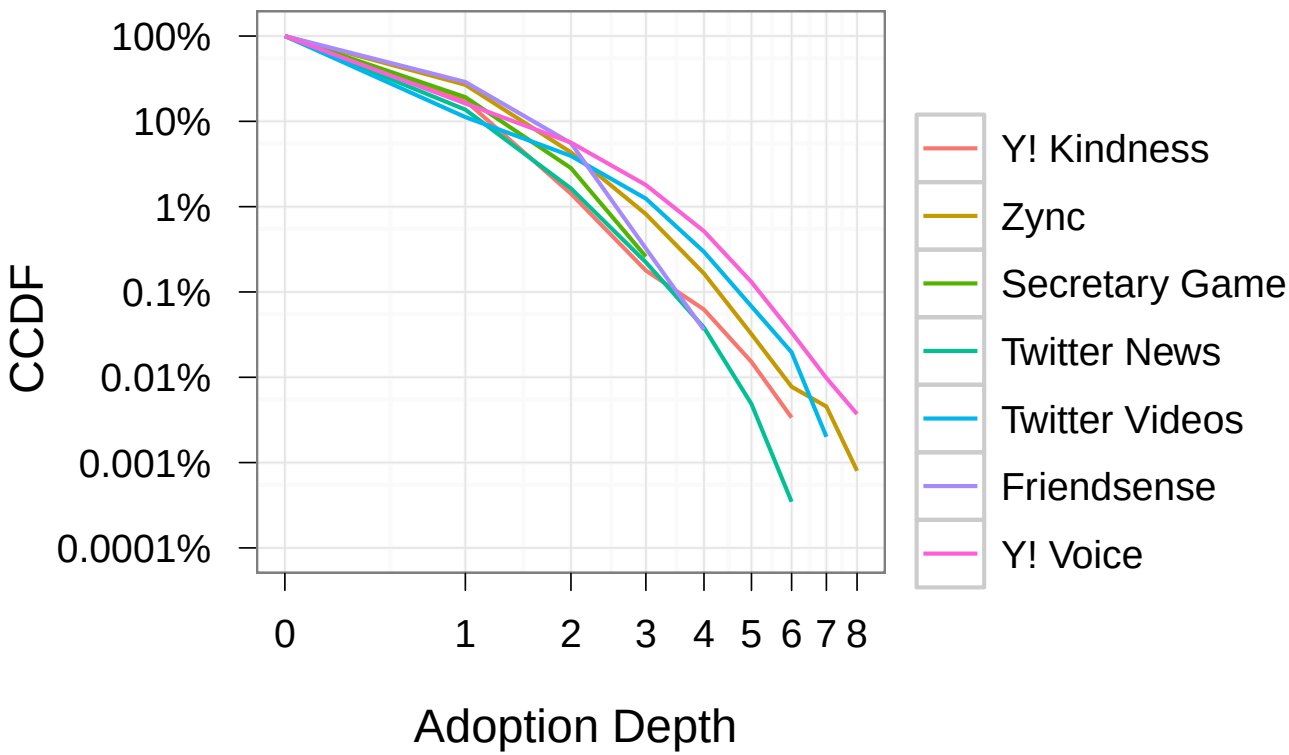
96% - 99% of trees die out within one generation

The finding that most trees are small and shallow is consistent with the intuition that size distributions are often right-skewed and heavy tailed.

The conventional wisdom is that a few huge trees are still dominating the diffusion process. For example, we could have:

- 99 single-node trees
- 1 huge, epidemic-like tree

In that case, the bulk of all adoption activity would still conform to our usual view of multi-step diffusion.



94% - 99% of adopters are within one generation of a seed

Mean Tree Size: 1.1 - 1.4

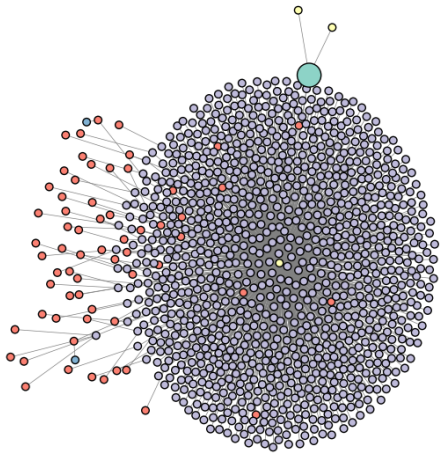
In the examples we study, "diffusion" is very well approximated by one-step propagation.

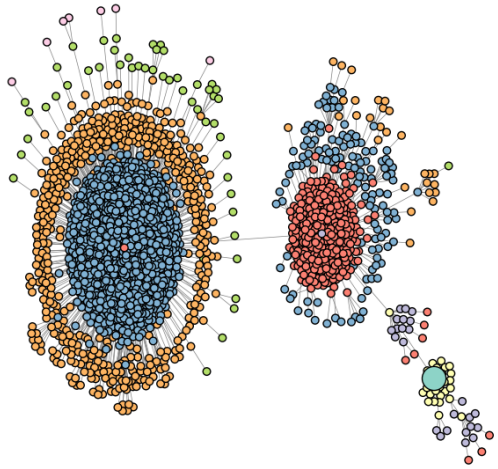
How general is this result?
What if all the products we study are just crappy?
(Note: they were in fact designed to be "viral")

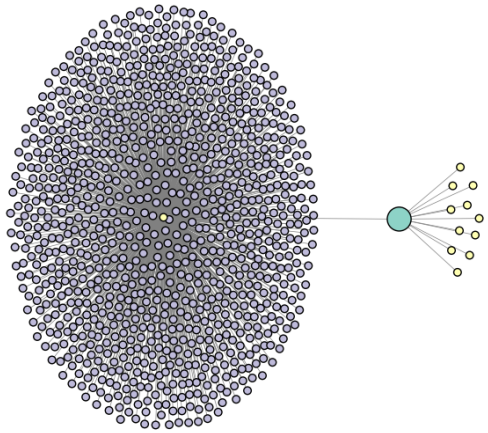
We consider the 16K Twitter links
that were independently introduced by at least 10 people

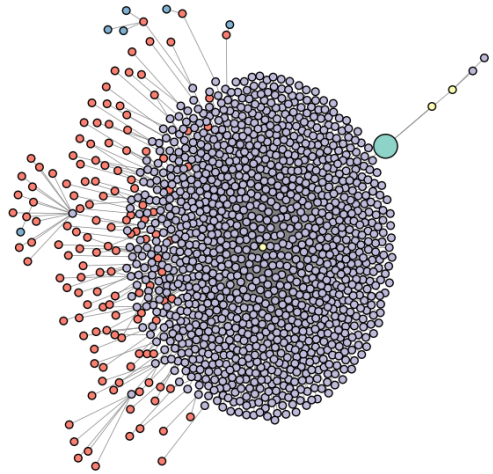
None of these "products" satisfy even a generous measure of "viral"
(i.e., having 90% of adoptions at least 2 steps away from the seed)

What about the rare, large events? Are those viral?





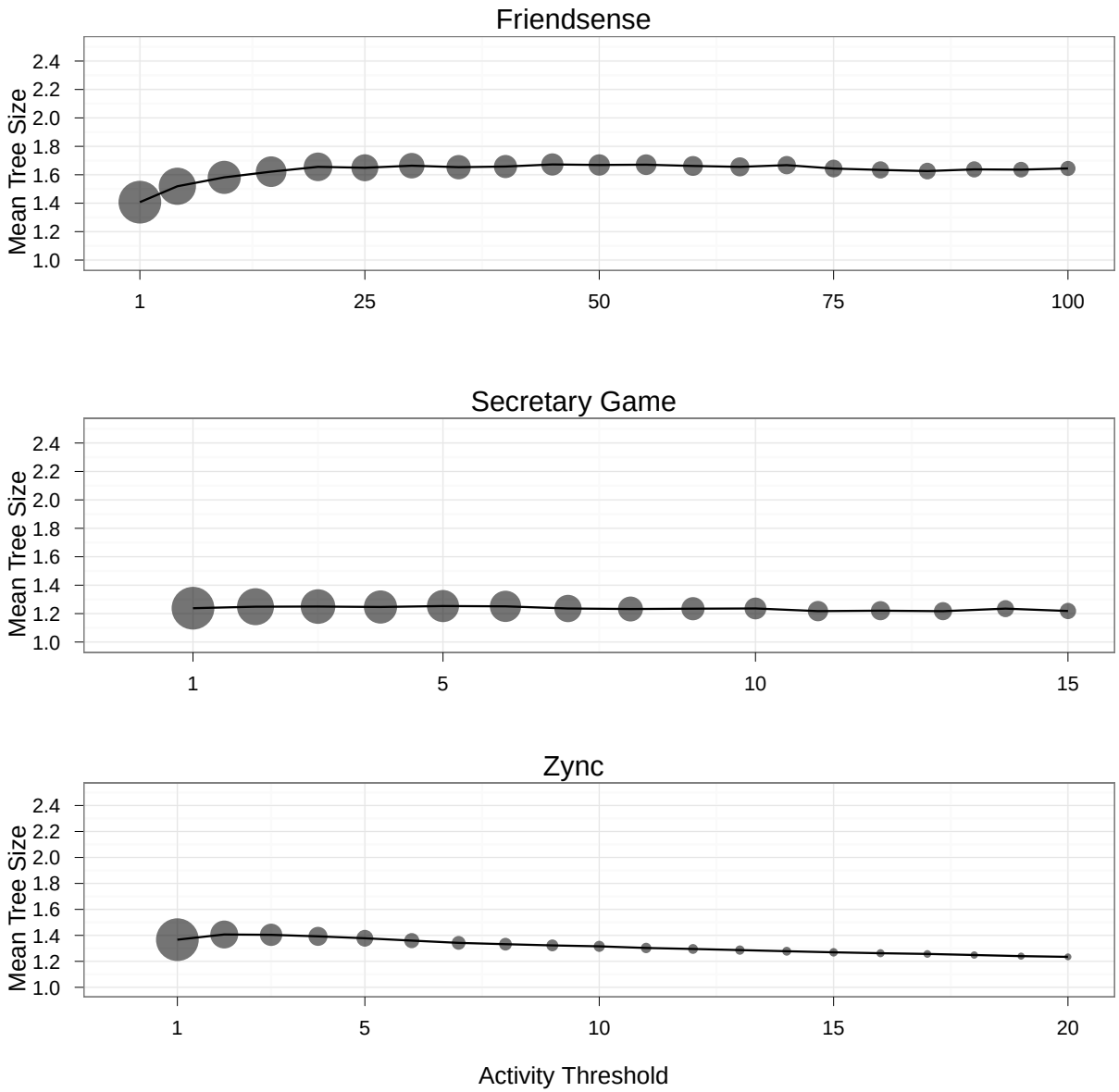




The structure of diffusion networks is consistent across the six domains we study, despite substantial differences in:

- the "product" begin diffused (e.g., URLs vs. a PC-to-phone service)
- the population of adopters
- the way in which we detect/infer peer-to-peer transmission

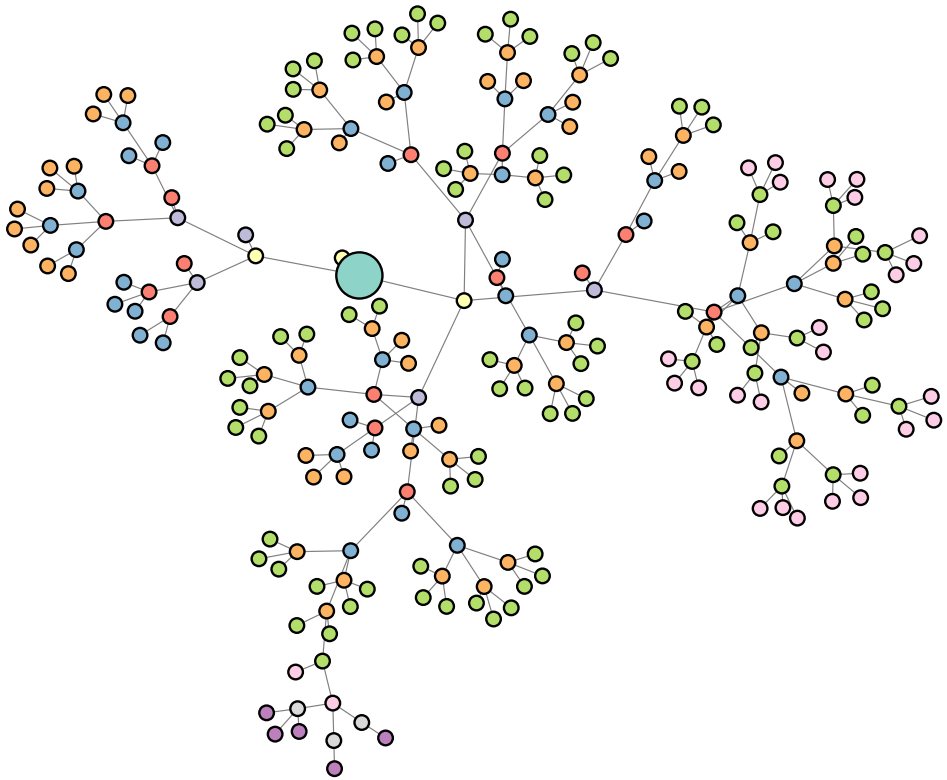
What if we vary the definition of "adoption" within a domain?


Friendsense
Mean Tree Size
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
1
25
50
75
100
Secretary Game
Mean Tree Size
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
1
5
10
15
Zync
Mean Tree Size
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
1
5
10
15
20
Activity Threshold

What about all the products that we "know" went viral? For example, YouTube hits and Hotmail.

Three possible answers:

1. They didn't actually "go viral" at all – driven by media or other "broadcast nodes"
2. Viral products have a key feature that's lacking in the domains we investigate (cf. the financial incentives of RDS)
3. They are precisely the rare events suggested by our data



Bastos, F.I., 2009

Implications

1. Diffusion of ideas is qualitatively different from the spread of disease. The viral analogy is not an accurate one.

2. "Viral boost" is smaller than generally believed; plan accordingly

3. Focus on taking mean tree size from 1.1 to 1.4