CSE 6242 A / CS 4803 DVA

Feb 19, 2013

Graphs I

Basics, how to build & store graphs, laws, etc.

Duen Horng (Polo) Chau

Georgia Tech

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Le Song

HW 1

Grades and feedback posted on T-Square

- Average score: 77 out of 80

Solution (SQL) posted on course website

HW2 out later this week

- Due after spring break
- You will have about a month to work on it

Graphs

Lecture 1 (today)

- Basics, how to build graph, store graph, laws, etc.

Lecture 2

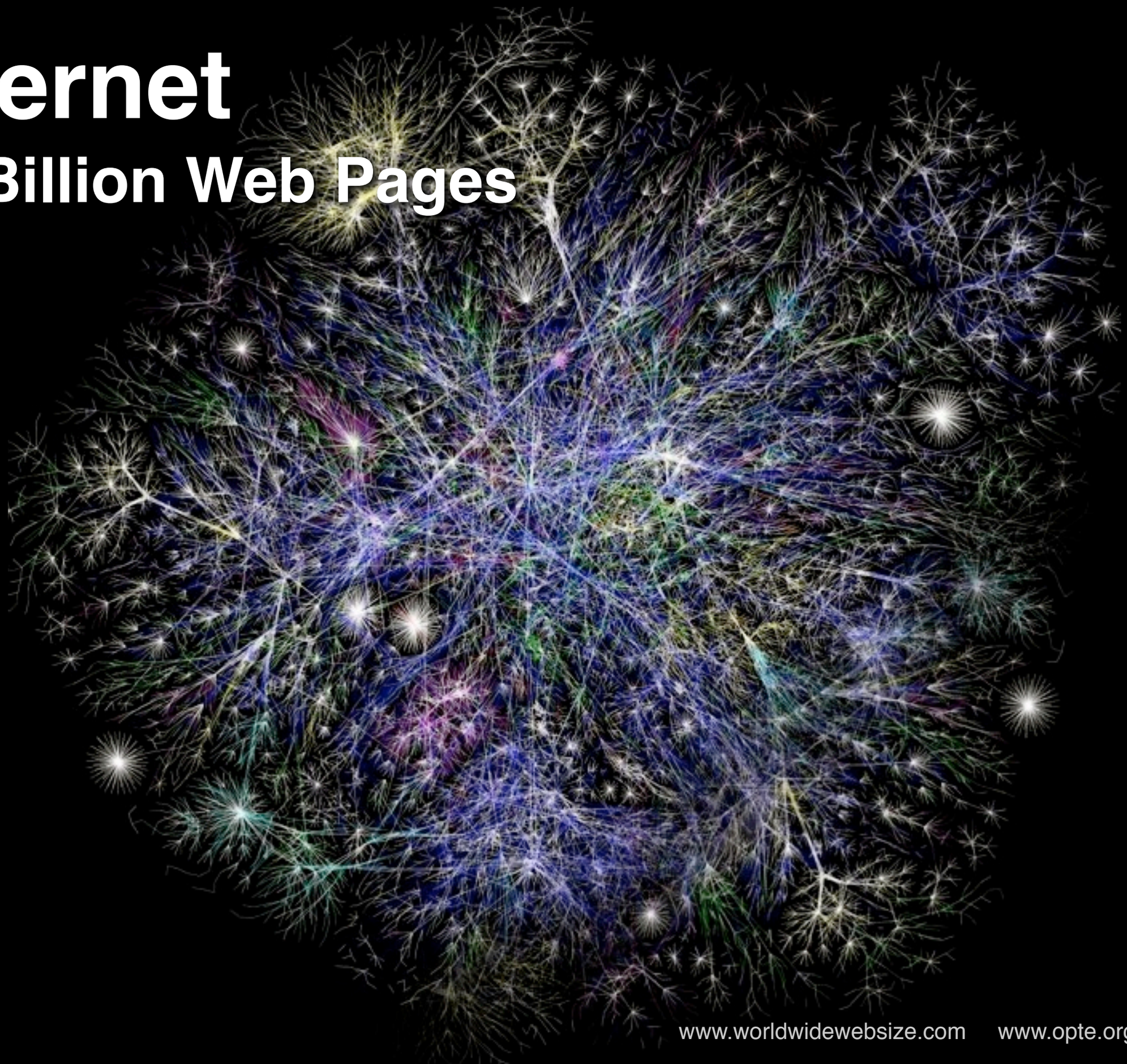
- centrality, scalable algorithms you need to know, how to visualize “large” graphs, challenges (research problems)

Lecture 3

- Interactive tools to make sense of large graphs, applications, etc.

Internet

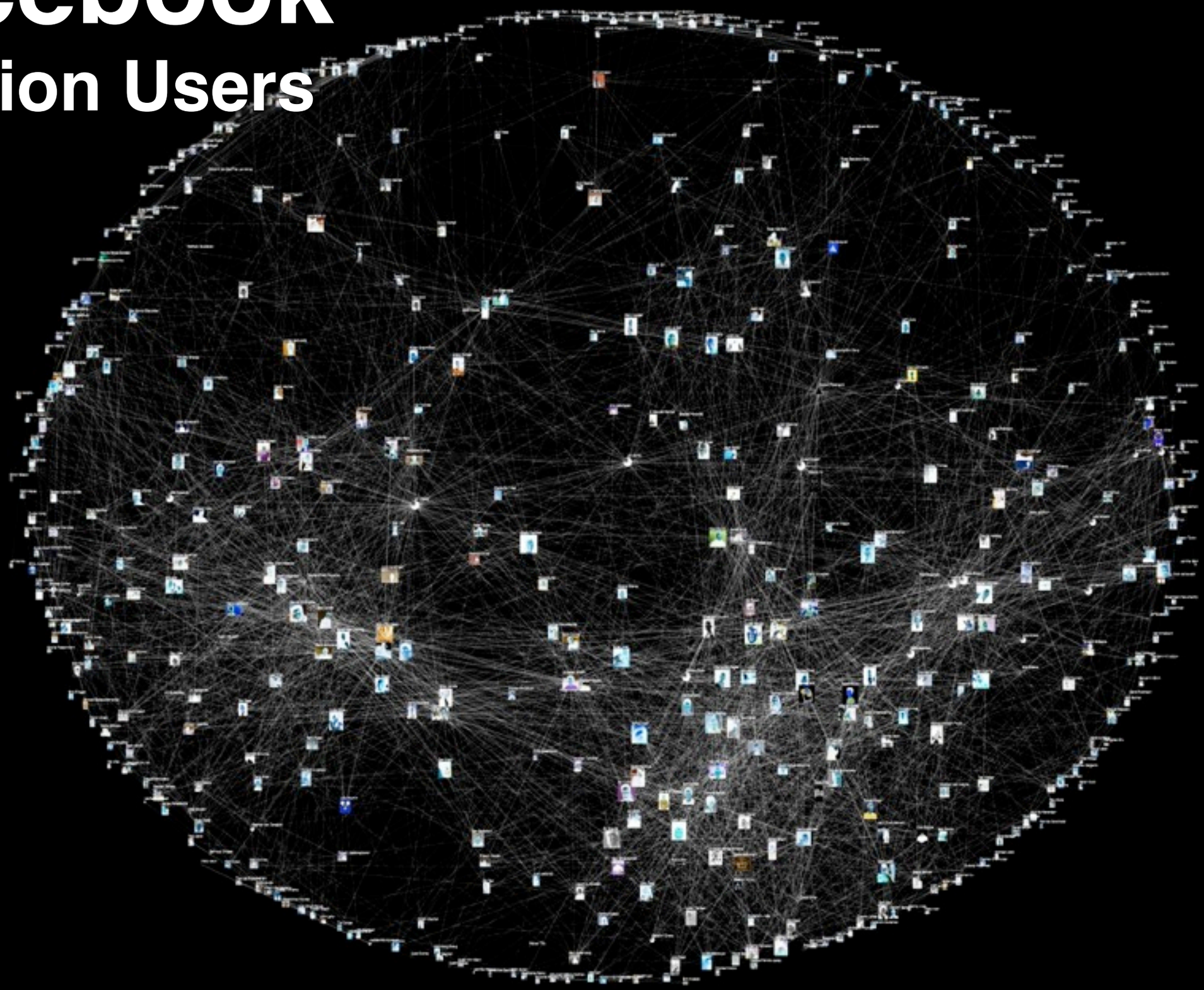
50 Billion Web Pages



www.worldwidewebsize.com www.opte.org

Facebook

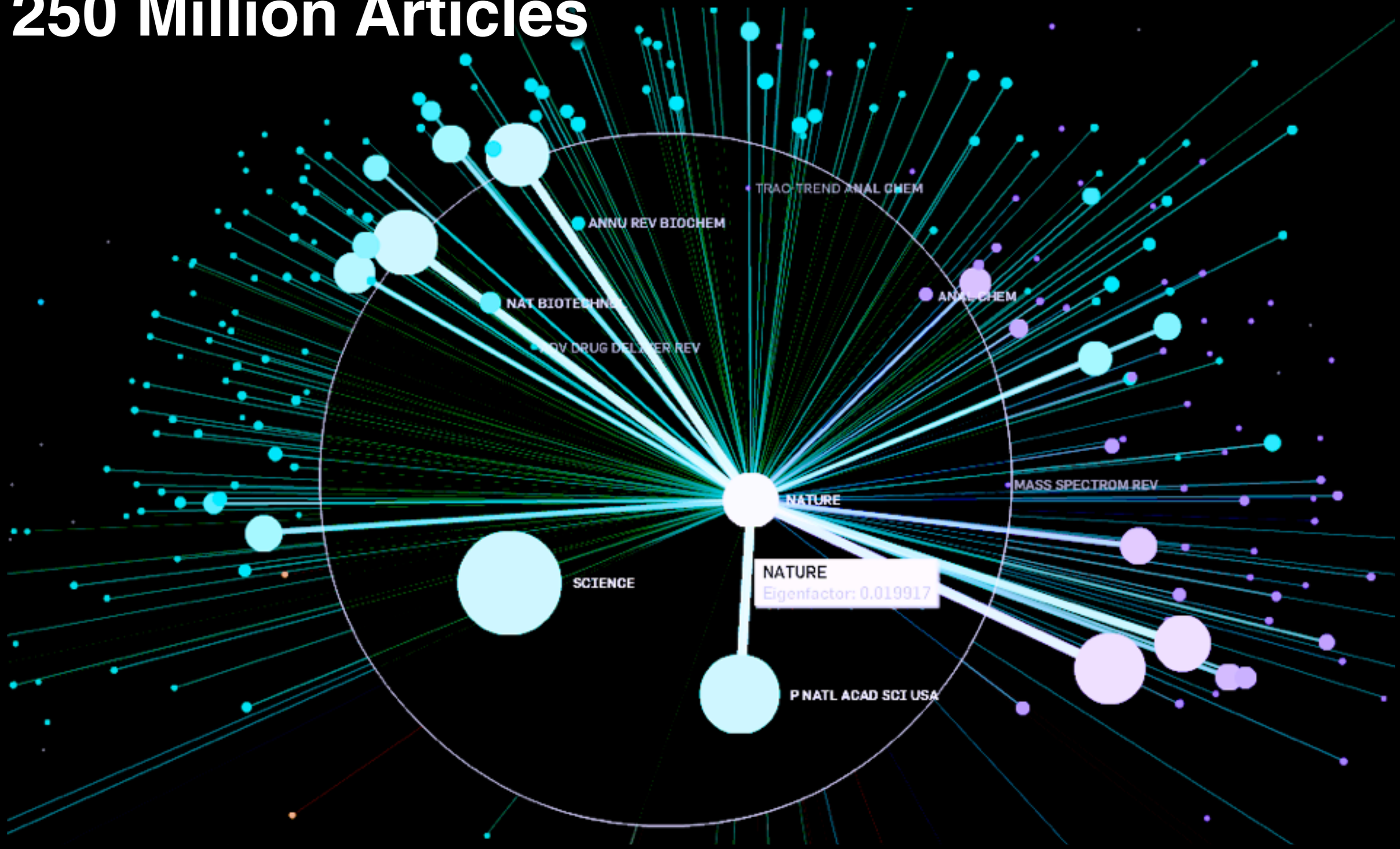
1 Billion Users



Modified from Marc_Smith, flickr

Citation Network

250 Million Articles



www.scirus.com/press/html/feb_2006.html#2 Modified from well-formed.eigenfactor.org

Many More

twitter

Who-follows-whom (500 million users)

amazon

Who-buys-what (120 million users)

at&t **cellphone network**

Who-calls-whom (100 million users)

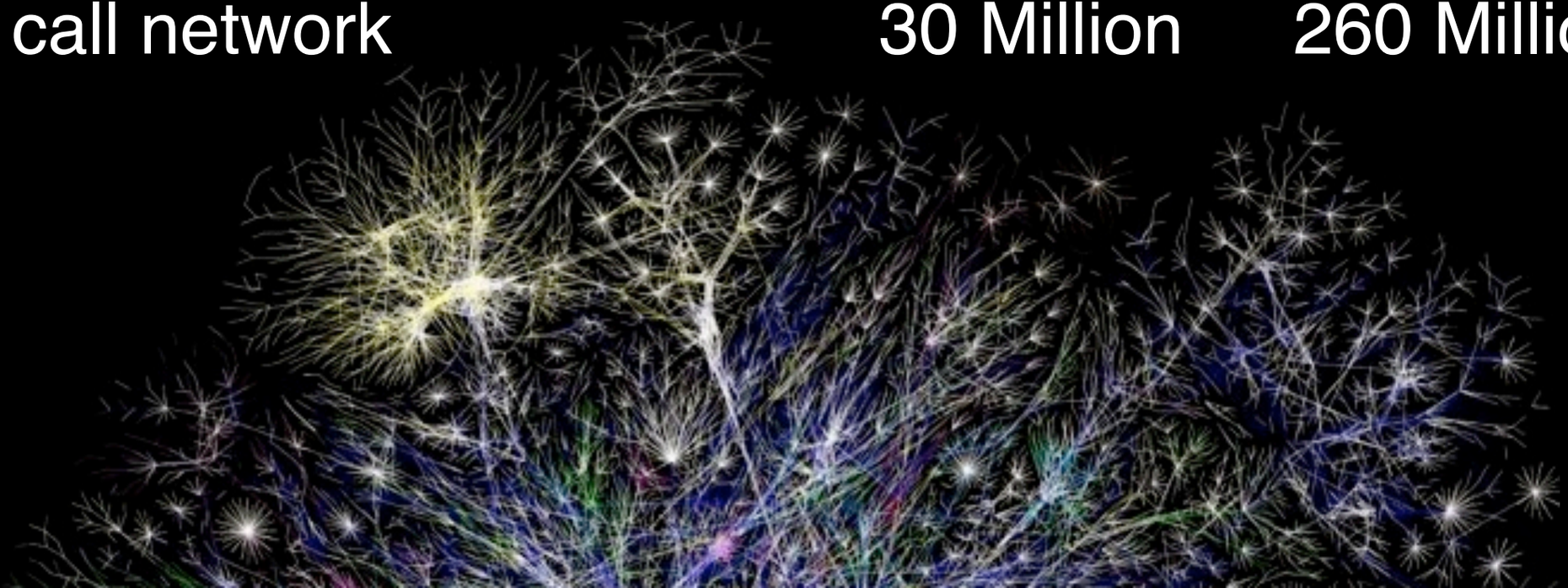
Protein-protein interactions

200 million possible interactions in human genome

Sources: www.selectscience.net www.phonedog.com www.mediabistro.com www.practicalecommerce.com/

Large Graphs I Analyzed

Graph	Nodes	Edges
YahooWeb	1.4 Billion	6 Billion
Symantec Machine-File Graph	1 Billion	**37 Billion**
Twitter	104 Million	3.7 Billion
Phone call network	30 Million	260 Million

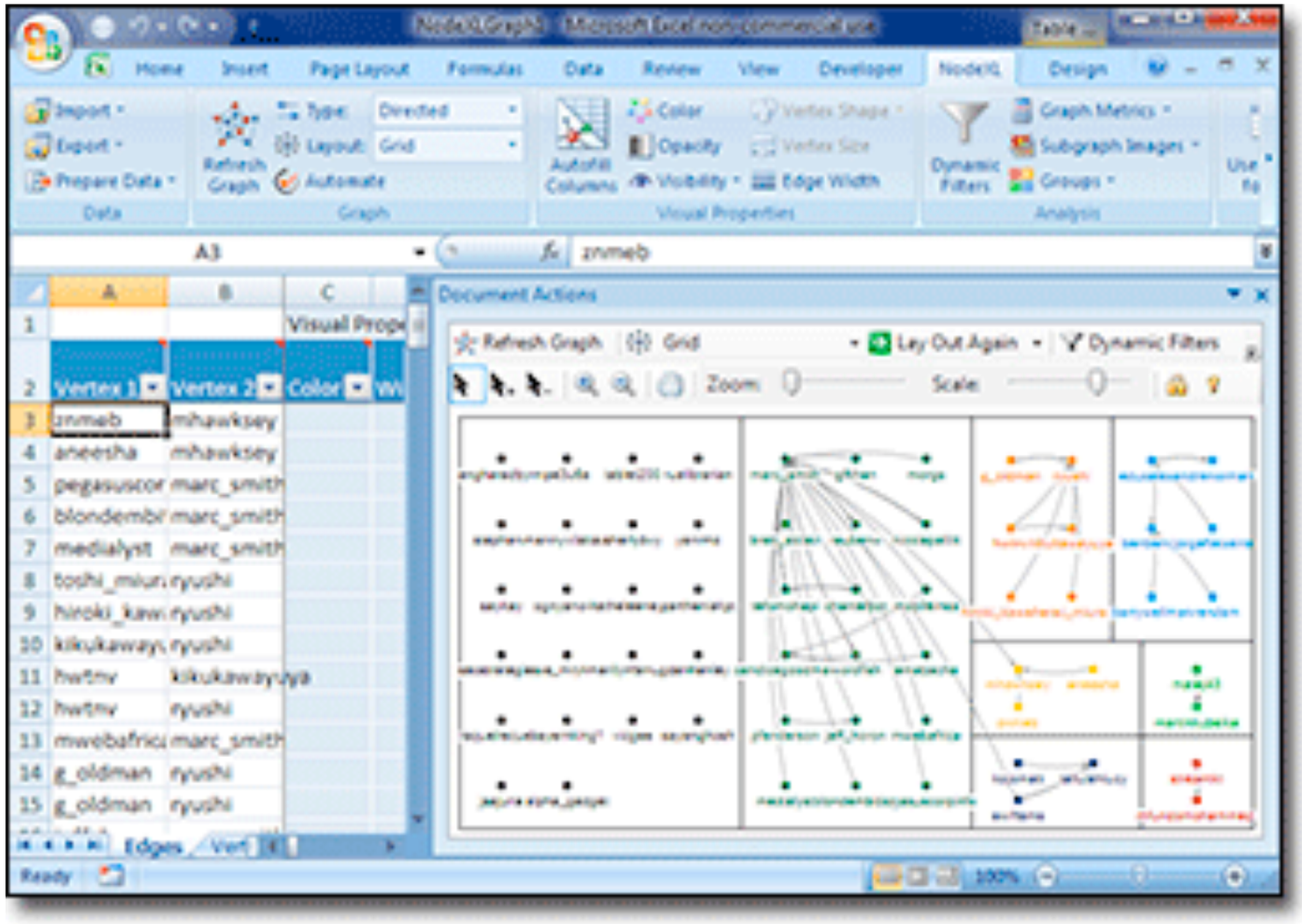


How to build a graph?

Use interactive tools

NodeXL http://nodexl.codeplex.com

- Excel plugin

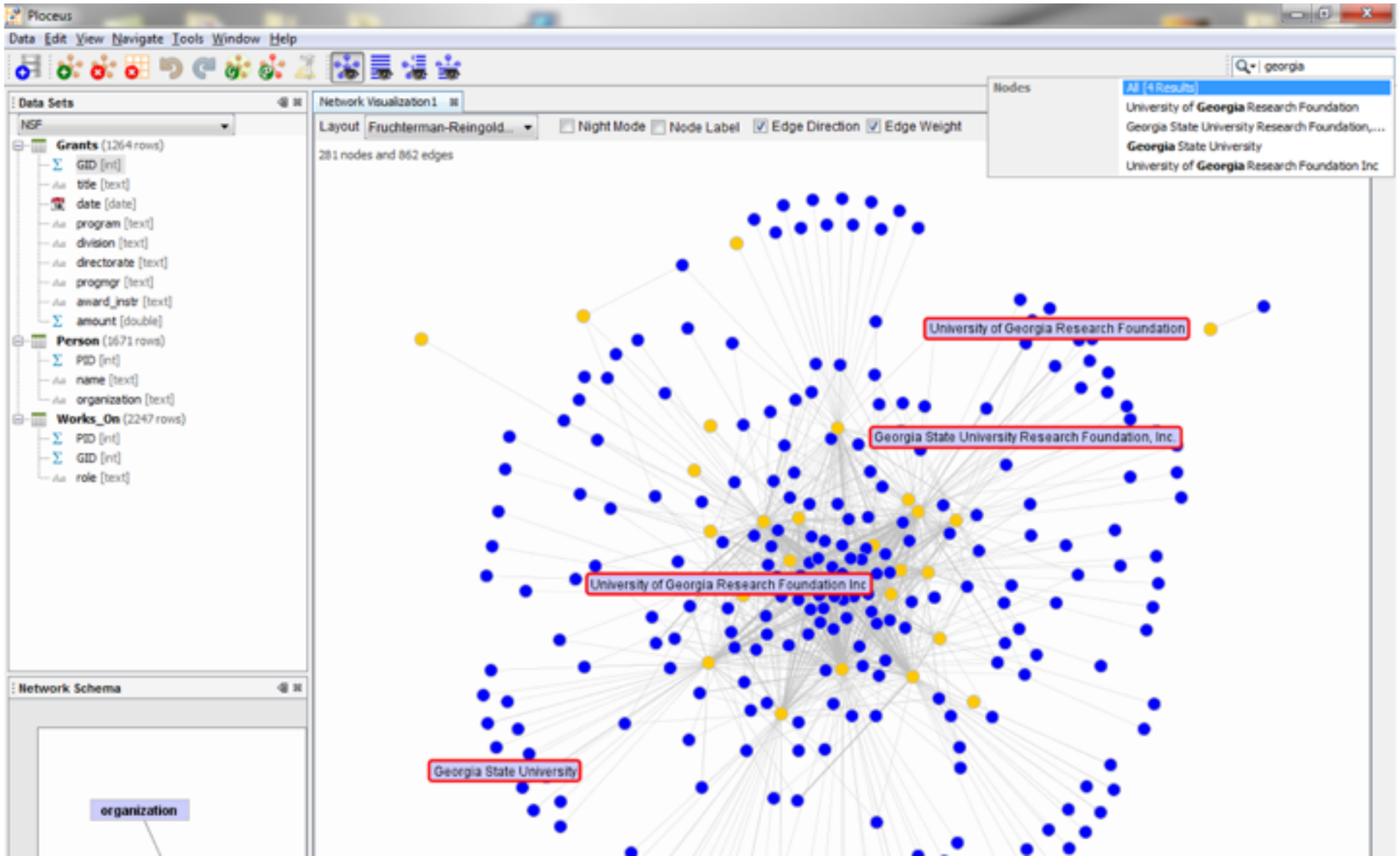


Use interactive tools

- http://www.cc.gatech.edu/gvu/ii/ploceus/

Ploceus: Network-based Visual Analysis of Tabular Data

- Zhicheng Liu, Sham Navathe, John Stasko. VAST 2011 (Made in Georgia Tech)



Slightly harder way: Use SQL

You already did this in HW1

- e.g., find pairs of actors/actresses who have starred in the same movie

How to store “large” graphs?

How large is “large”?

What do you think?

- In what units? Thousands? Millions?

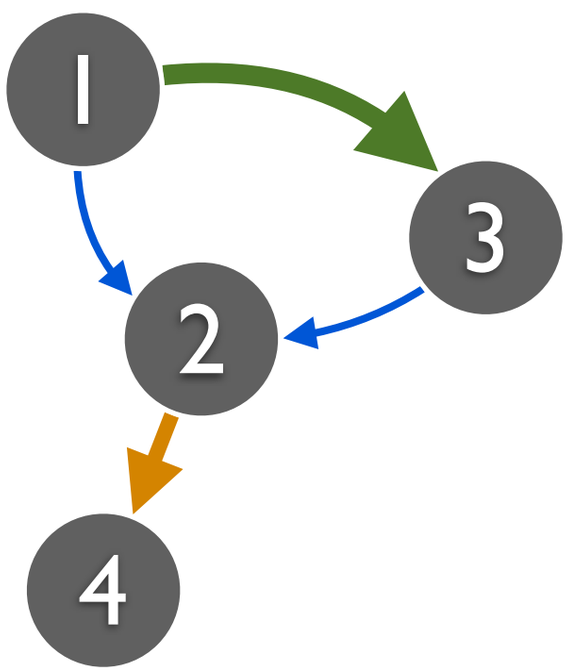
How do you measure a graph’s size?

- Such as...

Highly subjective. And domain specific.

How to represent a graph (and store it)?

Visually



Adjacency matrix

Target node

Source node

	1	2	3	4
1	0	1	3	0
2	0	0	0	2
3	0	1	0	0
4	0	0	0	0

Adjacency list

1: 2, 3
2: 4
3: 2

Edge list

1, 2, 1
1, 3, 3
2, 4, 2
3, 2, 1

- most common distribution format
- sometimes **painful** to parse when edges/nodes have many columns (some are text with double/single quotes, some are integers, some decimals, ...)

Storing large graphs...

On your laptop computer

- SQLite
- Neo4j (GPL license)

On a server

- MySQL, PostgreSQL, etc.
- Neo4j(?)

With a cluster (more details a few lectures down)

- **Hadoop** (generic framework)
- **HBase**(?) , inspired by Google's BigTable
- **Hama**, inspired by Google's Pregel
- **FlockDB**, by Twitter
- Comparison of "graph databases"
  http://nosql.mypopescu.com/post/40759505554/a-comparison-of-7-graph-databases

Storing large graphs on your computer

I like to use **SQLite**. Why?

- Easily handle up to **gigabytes**
  - Roughly **tens of millions** of nodes/edges (perhaps up to billions?). Very good! For **today's** standard.
- Very easy to maintain: **one** cross-platform file
- Has programming wrappers in numerous languages
  - C++, Java (Andriod), Python, Objective C (iOS),...
- Queries are so easy!
  e.g., find all nodes' degrees = 1 SQL statement
- Bonus: SQLite even supports full-text search

SQLite graph database schema

Simplest schema:

```
edges(source_id, target_id)
```

More sophisticated (flexible; lets you store more things):

```
CREATE TABLE nodes (
    id INTEGER PRIMARY KEY,
    type INTEGER DEFAULT 0,
    name VARCHAR DEFAULT '');

CREATE TABLE edges (
    source_id INTEGER,
    target_id INTEGER,
    type INTEGER DEFAULT 0,
    weight FLOAT DEFAULT 1,
    timestamp INTEGER DEFAULT 0,
    PRIMARY KEY(source_id, target_id, timestamp));
```

Side note:

Full-Text Search (FTS) on SQLite

http://www.sqlite.org/fts3.html

Very simple. Built-in. Only needs 3 lines of commands.

- **Create** FTS table (index)

```
CREATE VIRTUAL TABLE critics_consensus USING
fts4(consensus);
```

- **Insert** text into FTS table

```
INSERT INTO critics_consensus SELECT
critics_consensus FROM movies;
```

- **Query** using the “match” keyword

```
SELECT * FROM critics_consensus WHERE consensus MATCH
'funny OR horror';
```

Originally developed by Google engineers

Project idea

- Compare scalability between SQLite, Neo4j, HBase, etc.
- Which uses more space? What's the maximum graph size?
- Which answers queries the fastest? For what queries? How does that change with the graph size?

Related-Movie Graph

On T-Square, under “Resources”. **Don’t distribute!**

Thanks to Mr. Aakash Goel, for building the crawler (still crawling)

127,703 actors

- id, name

118,431 movies (nodes)

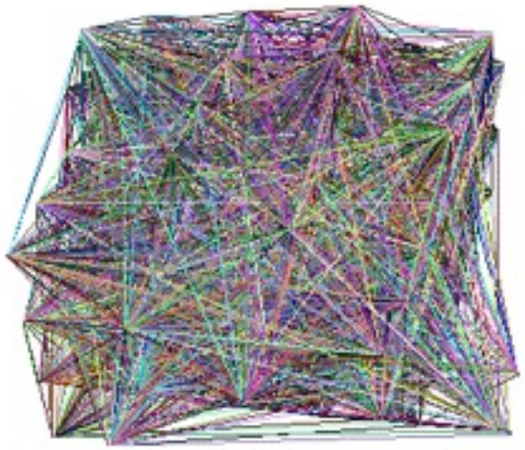
- many attributes: id, title, year, genres, etc.

16,856 pairs of related movies (edges)

- this means many movies are “singletons” (without any related movies)

131 MB

I have a graph dataset. Now what?



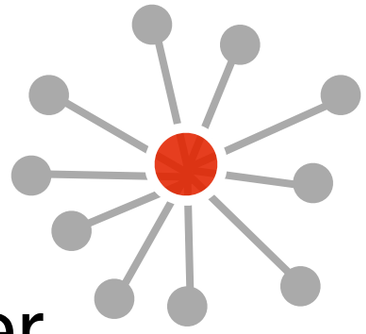
Analyze it! Do "**data mining**" or "**graph mining**".

How does it "look like"? Visualize it if it's small.

Yuck.

Does it follow any expected patterns?
Or does it *not* follow some patterns (outliers)?

- Why does this matter?
- If we know the **patterns** (models), we can do **prediction**, **recommendation**, etc.
  e.g., is Alice going to "friend" Bob on Facebook?
  People often buy beer and diapers together.
- **Outliers** often give us **new insights**
  e.g., telemarketer's friends don't know each other



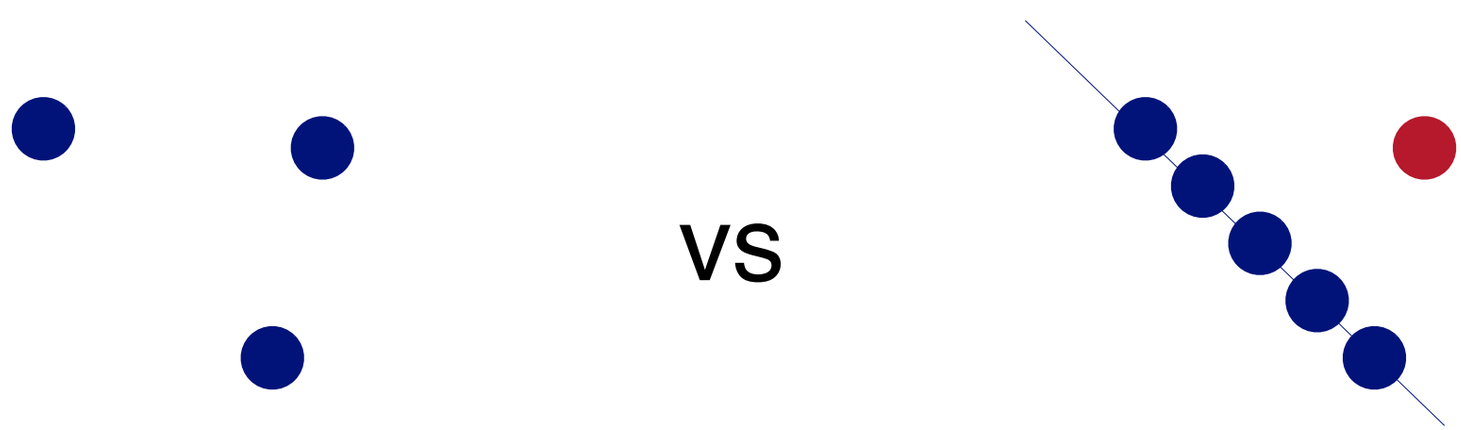
Finding patterns & outliers in graphs

Outlier/Anomaly detection (will be covered later)

- To spot them, we need to patterns first
- Anomalies = things that do not fit the patterns

To effectively do this, we need large datasets

- patterns and anomalies don’t show up well in small datasets



Are real graphs random?

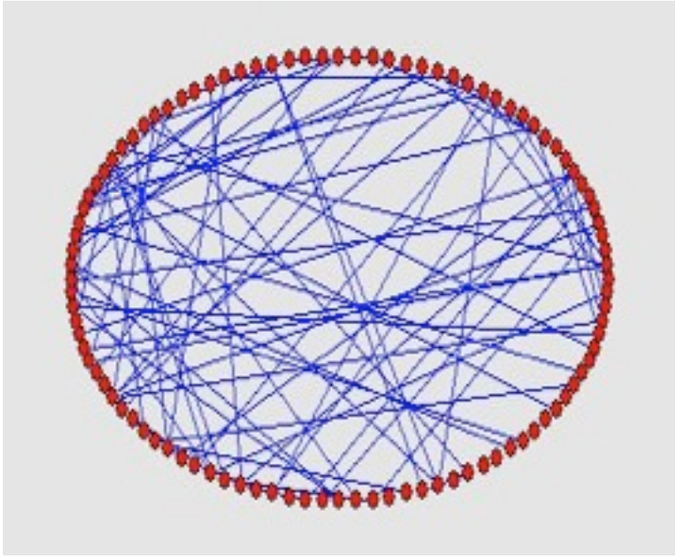
Random graph (Erdos-Renyi)
100 nodes, avg degree = 2

No obvious patterns

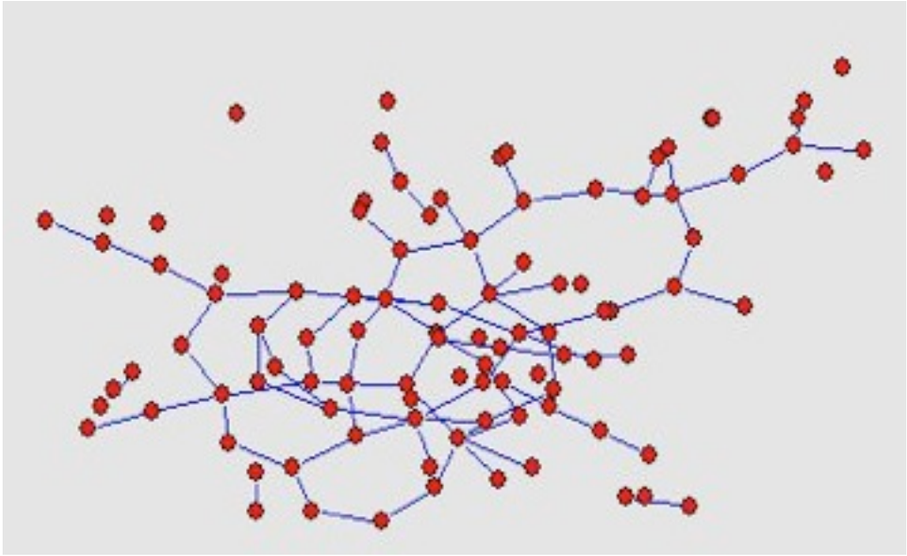
Generated with pajek

http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Before layout



After layout



Graph mining

- Are real graphs random?

Laws and patterns

- Are real graphs random?
- A: NO!!
  - Diameter (longest shortest path)
  - in- and out- degree distributions
  - other (surprising) patterns

- So, let's look at the data

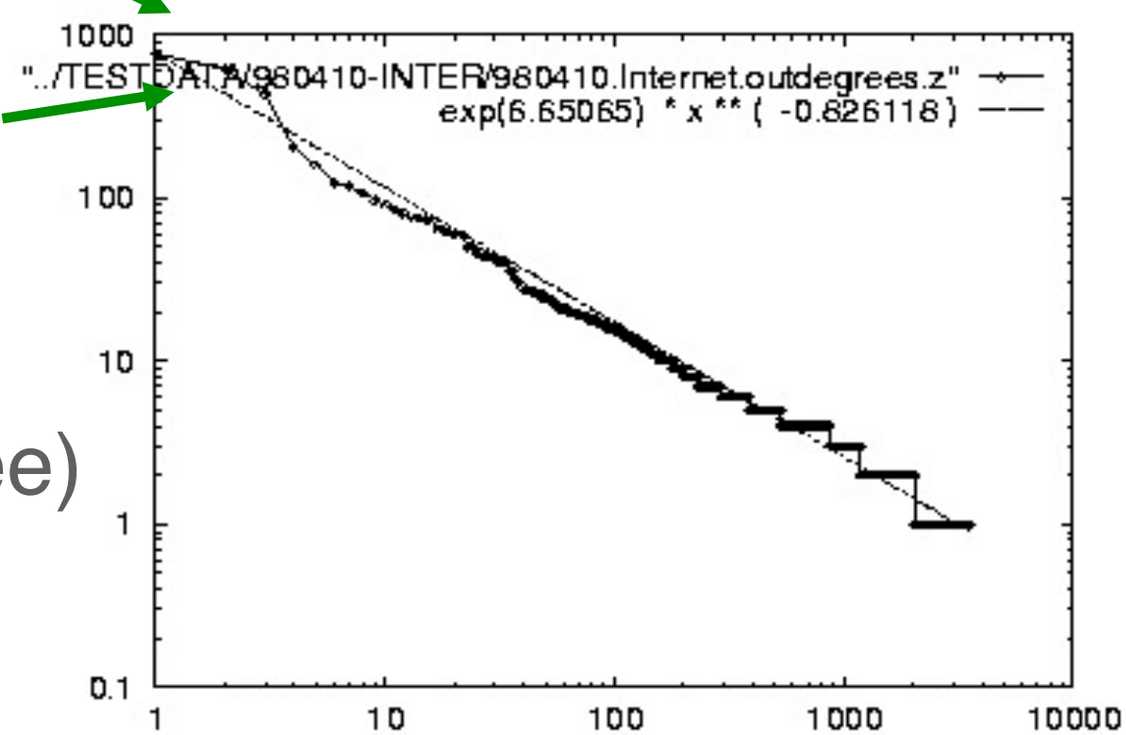
Power Law in Degree Distribution

- Faloutsos, Faloutsos, Faloutsos [SIGCOMM99] Seminal paper. Must read!

internet domains

att.com

ibm.com

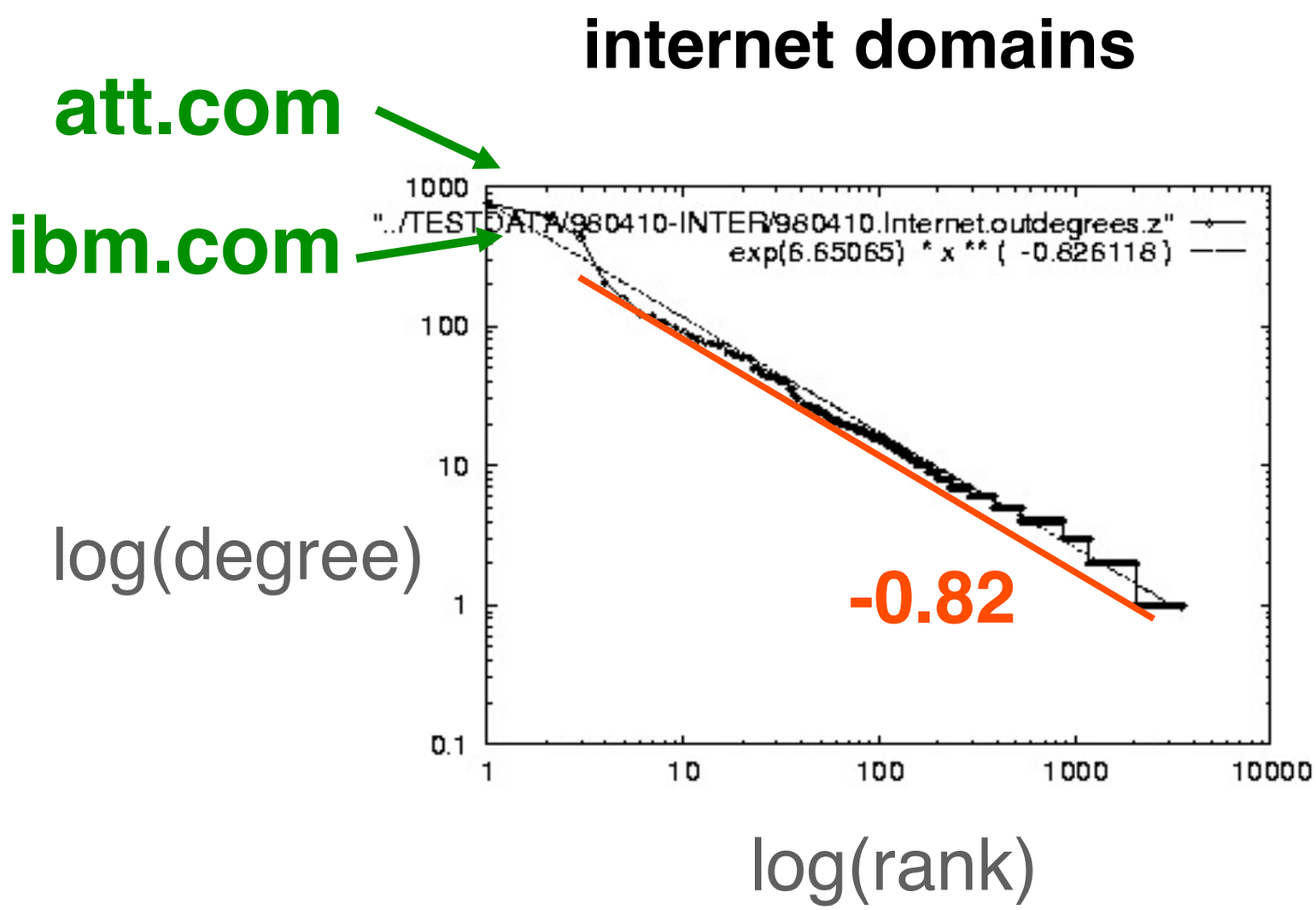


log(degree)

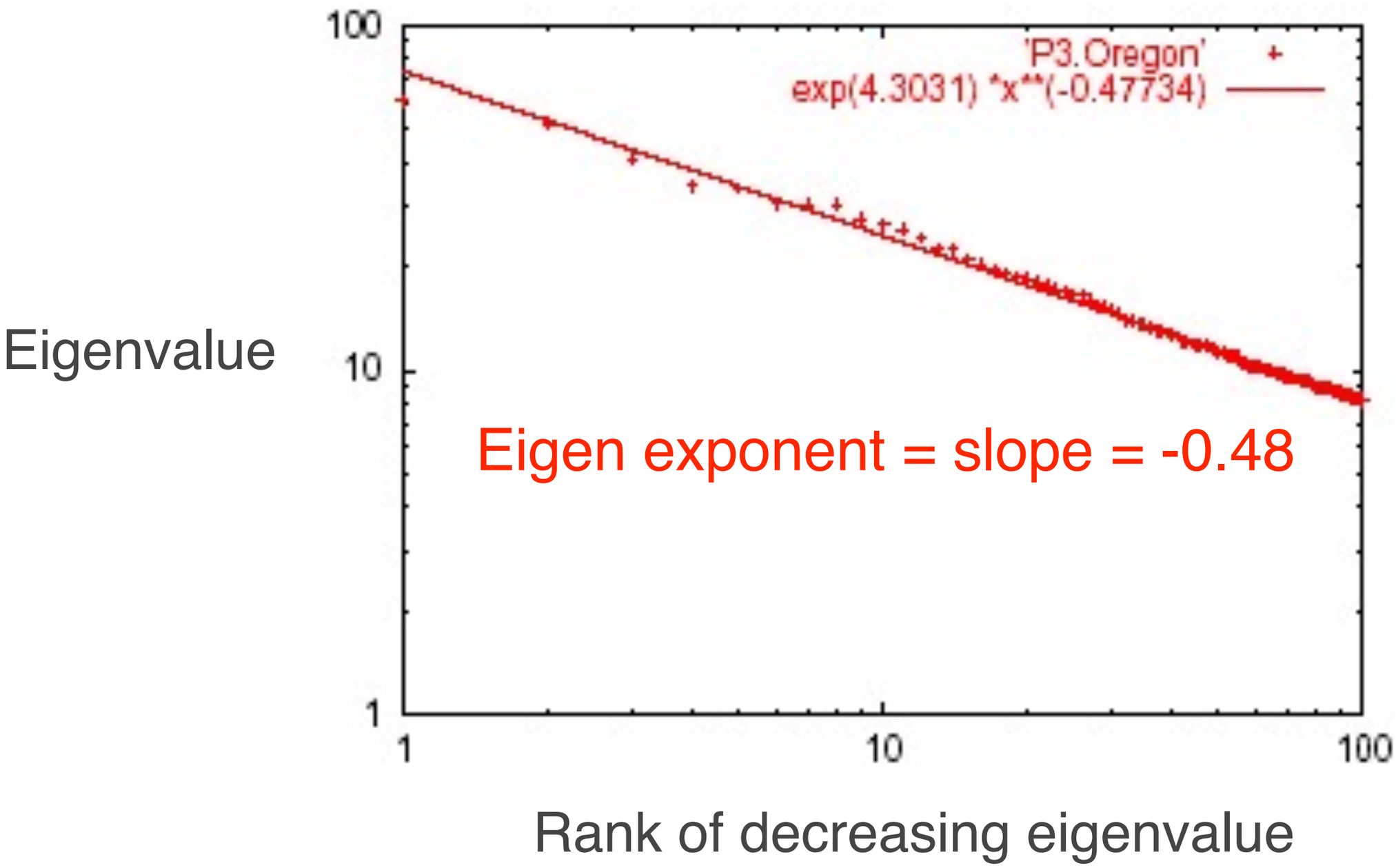
log(rank)

Power Law in Degree Distribution

- Faloutsos, Faloutsos, Faloutsos [SIGCOMM99] Seminal paper. Must read!



Power Law in Eigenvalues of Adjacency Matrix



How about graphs from other domains?

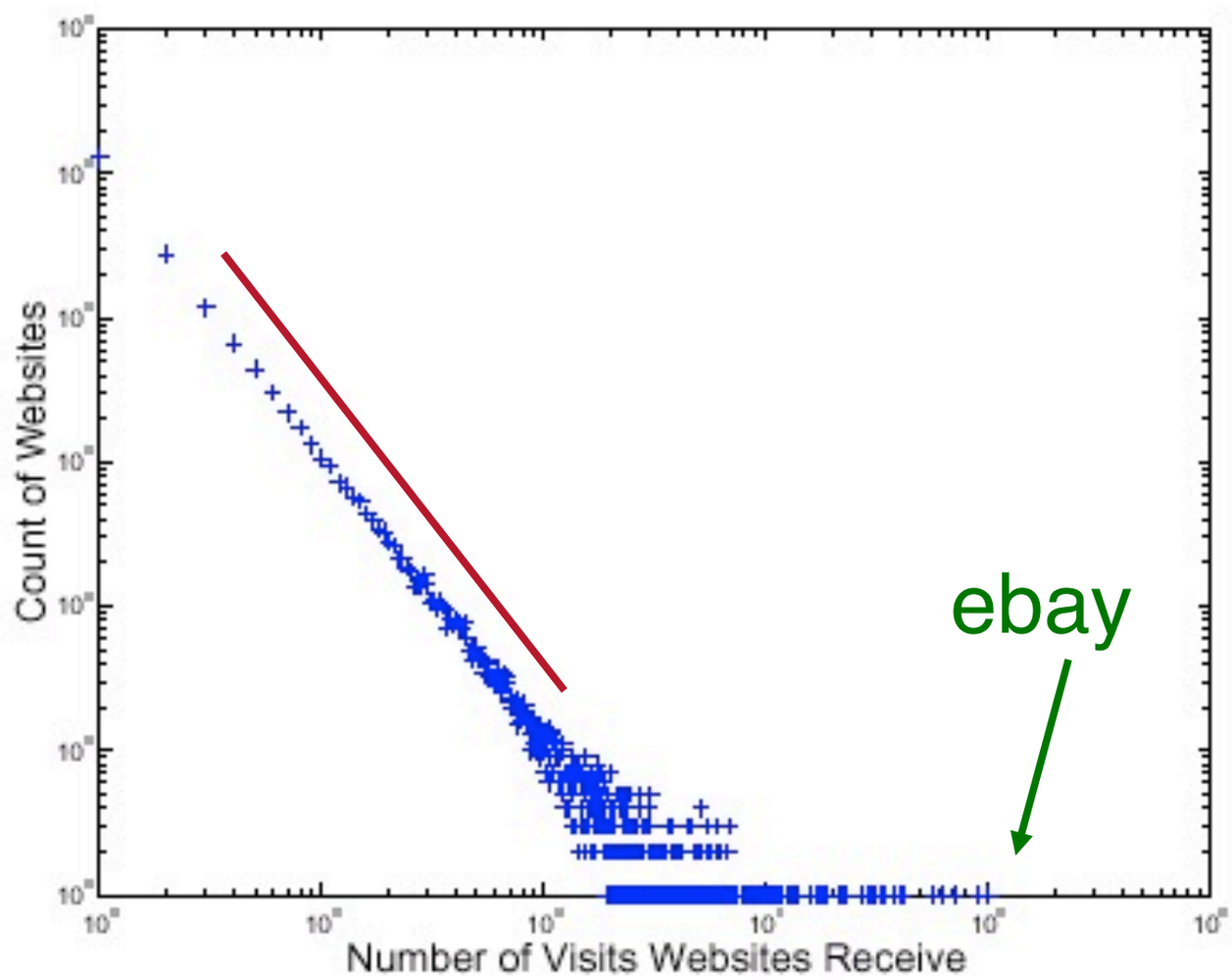
More Power Laws

- Web hit counts

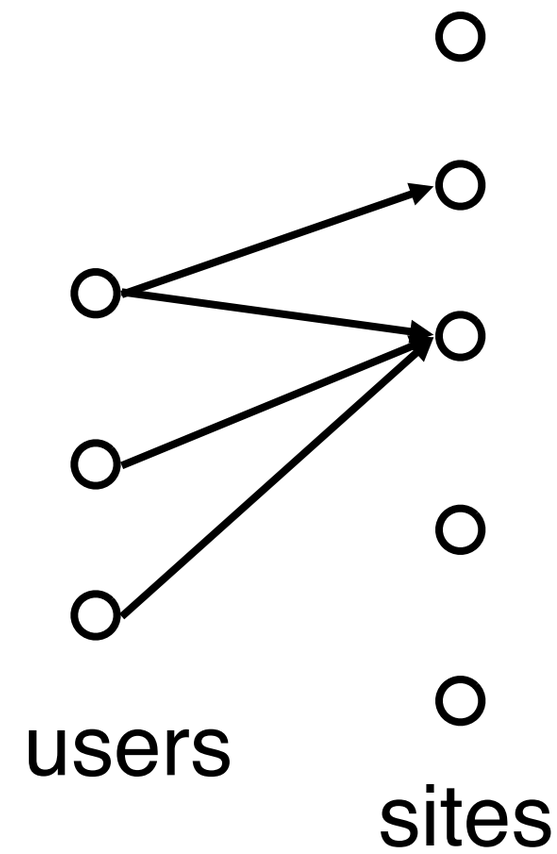
  [Alan L. Montgomery and Christos Faloutsos]

Web Site Traffic

log(#website)



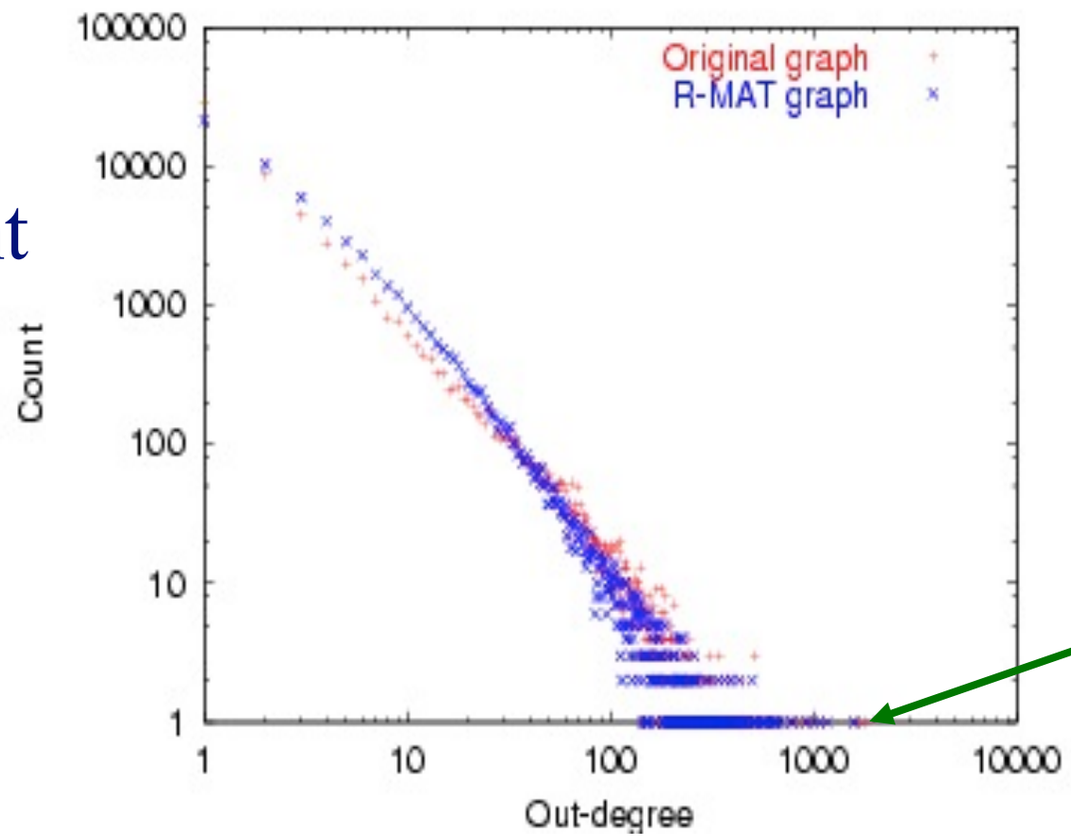
log(#website visit)



epinions.com

- who-trusts-whom [Richardson + Domingos, KDD 2001]

count



trusts-2000-people user

(out) degree

And numerous more

- # of sexual contacts
- Income [Pareto] – 80-20 distribution
- Duration of downloads [Bestavros+]
- Duration of UNIX jobs
- File sizes
- …

Any other ‘laws’?

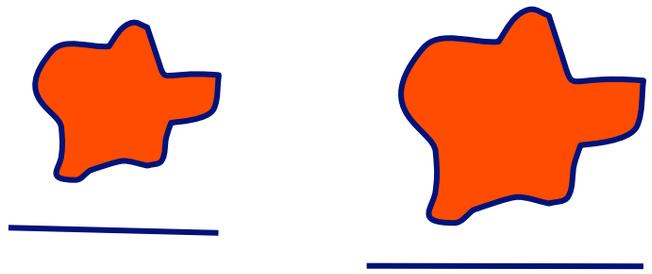
- Yes!
- Small diameter (~ constant!) –
  - six degrees of separation / ‘Kevin Bacon’
  - small worlds [Watts and Strogatz]

Problem: Time evolution

- Jure Leskovec (CMU -> Stanford)
- Jon Kleinberg (Cornell)
- Christos Faloutsos (CMU)

Evolution of the Diameter

- Prior work on Power Law graphs hints at slowly growing diameter:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?



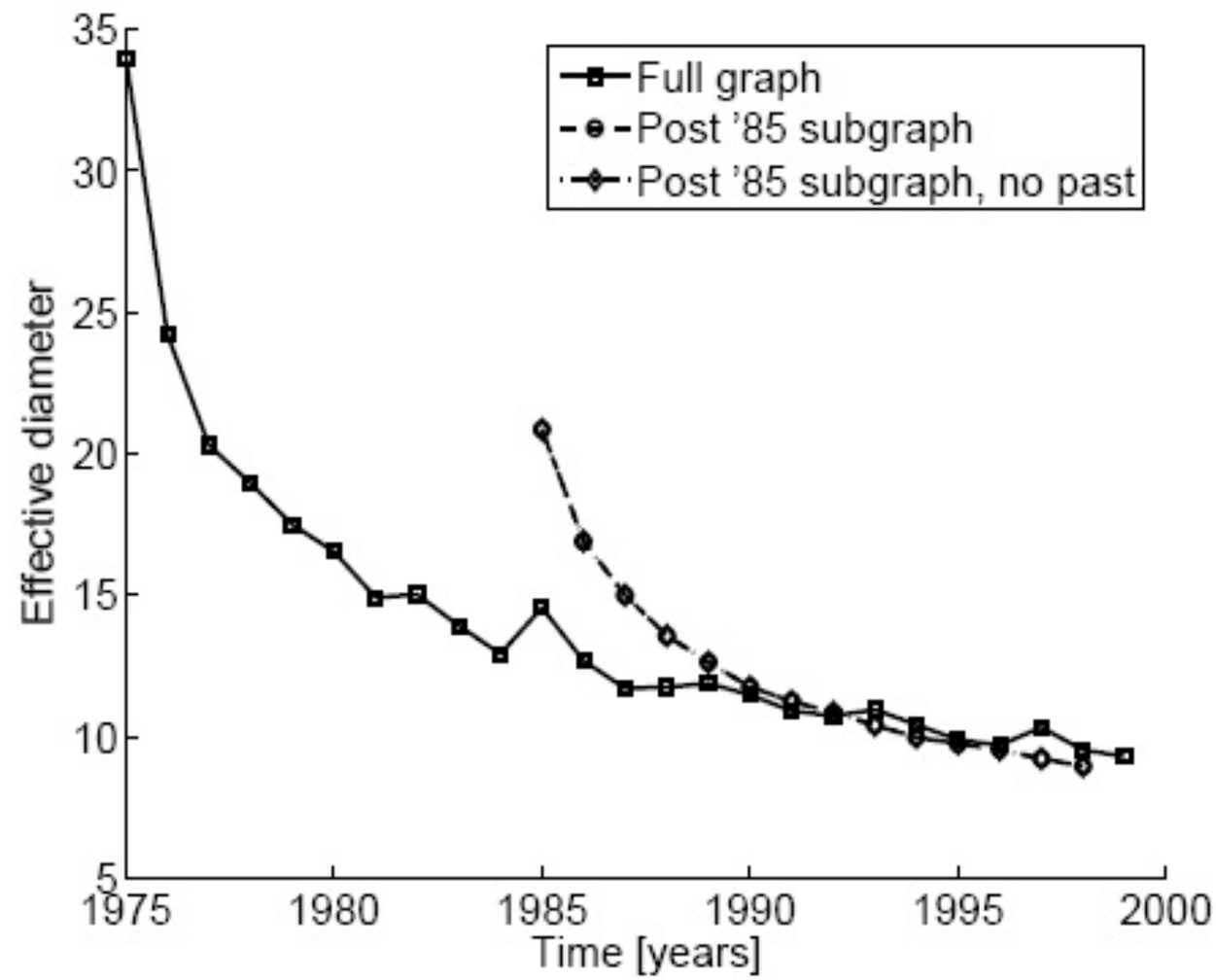
Evolution of the Diameter

- Prior work on Power Law graphs hints at slowly growing diameter:
  - diameter ~ O(log N)
  - diameter ~ O(log log N)
- What is happening in real data?
- Diameter shrinks over time

Diameter – “Patents”

- Patent citation network
- 25 years of data
- @1999
  - 2.9 M nodes
  - 16.5 M edges

diameter



time [years]

Temporal Evolution of the Graphs

- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that
  - $$N(t+1) = 2 * N(t)$$
- Q: what is your guess for
  - $$E(t+1) =? 2 * E(t)$$

Temporal Evolution of the Graphs

- N(t) … nodes at time t
- E(t) … edges at time t
- Suppose that

$$N(t+1) = 2 * N(t)$$

- Q: what is your guess for

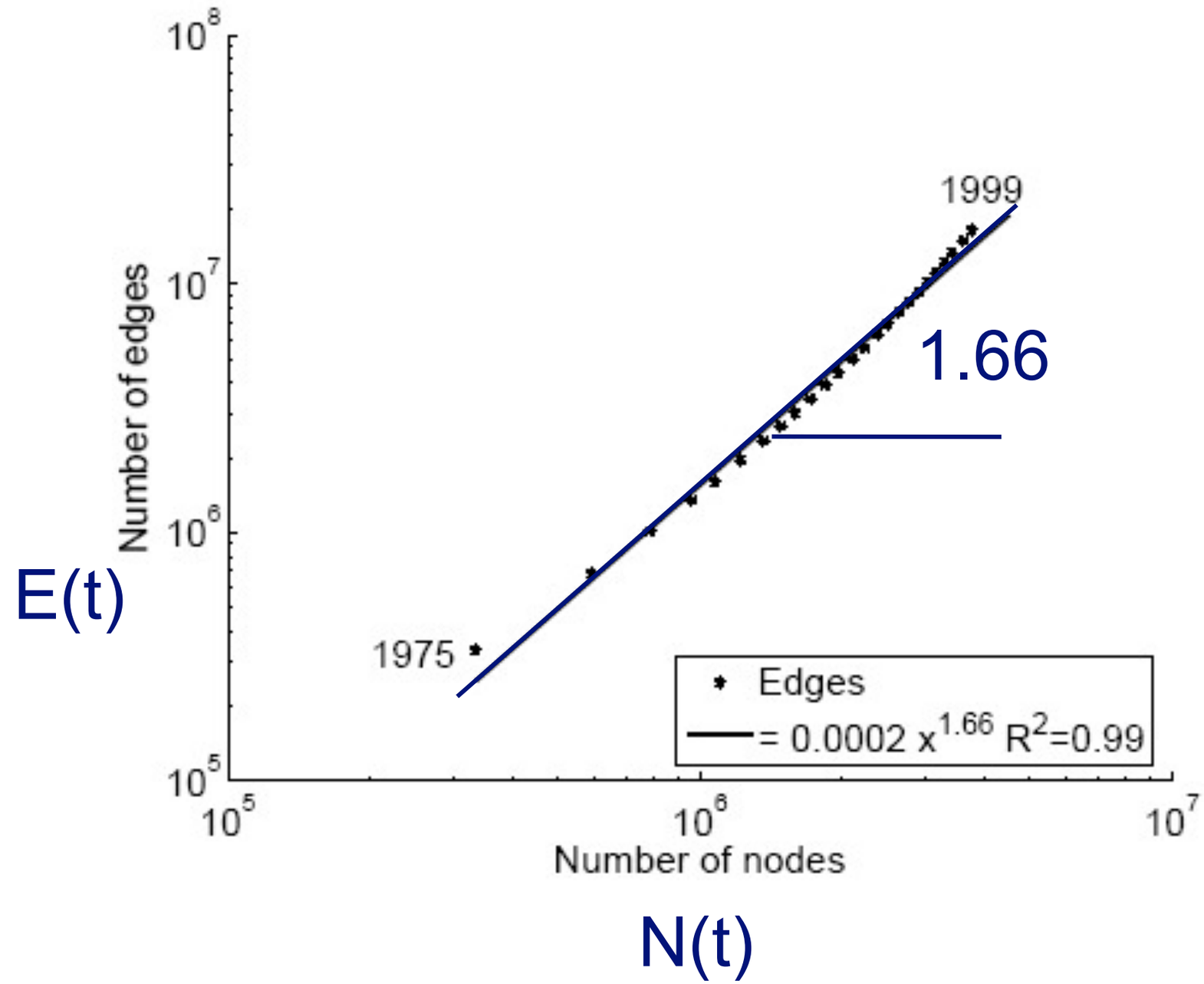
$$E(t+1) =? \; 2 * E(t)$$

- A: over-doubled!

  But obeying the ``Densification Power Law”

Densification – Patent Citations

- Citations among patents granted
- @1999
  - 2.9 M nodes
  - 16.5 M edges
- Each year is a datapoint



So many laws!

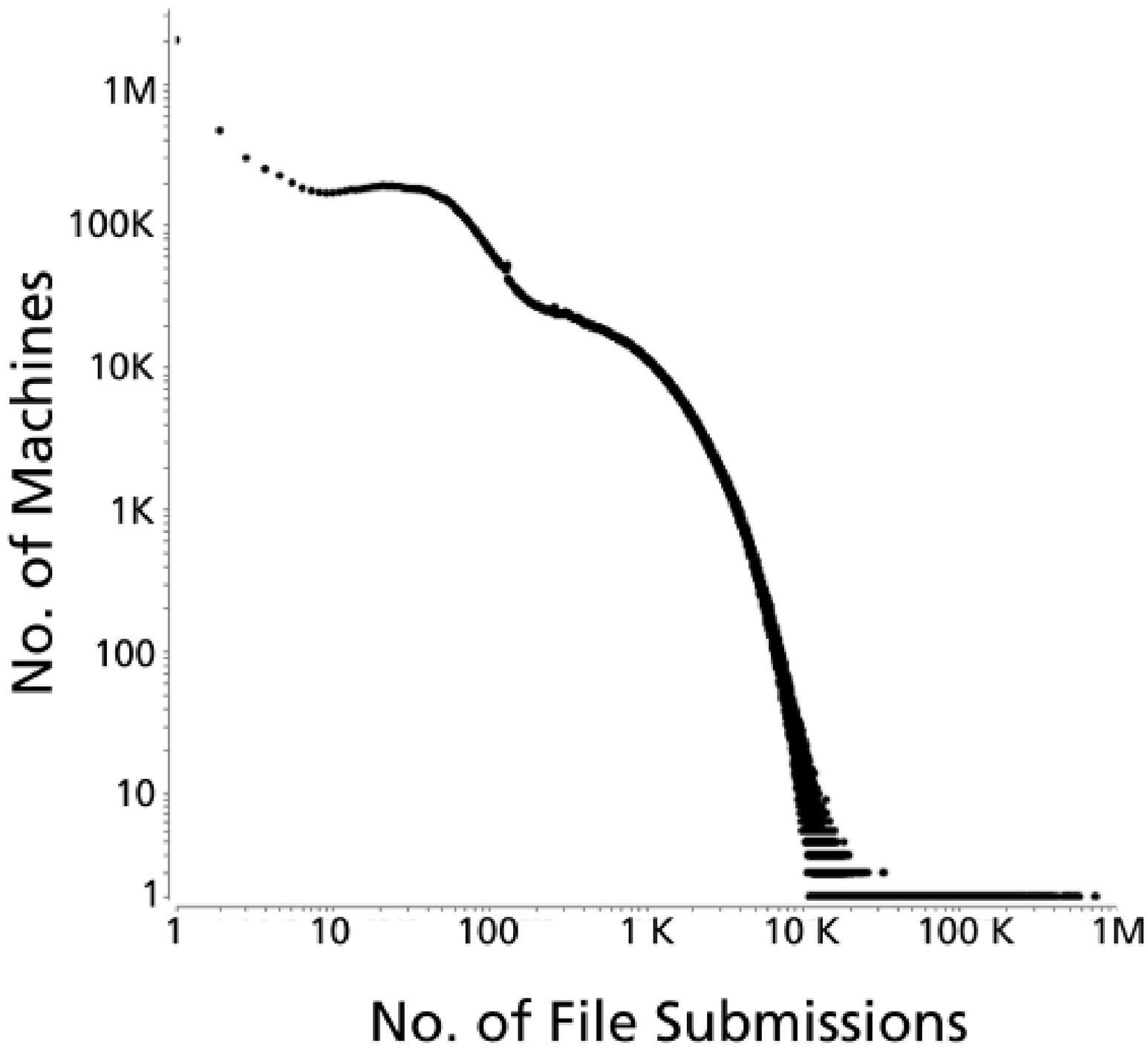
There will be more to come...

To date, there are 11 (or more) laws

- RTG: A Recursive Realistic Graph Generator using Random Typing [Akoglu, Faloutsos]

What should you do?

- Try as many distributions as possible and see if your graph fits them.
- If it doesn't, find out the reasons. Sometimes it's due to errors/problems in the data; sometimes, it signifies some new patterns!



Polonium: Tera-Scale Graph Mining and Inference for Malware Detection [Chau, et al]