UNIVERSITY OF CAMBRIDGE
800 YEARS
1209 ~ 2009

# Eight Friends are Enough: Social Graph Approximation via Public Listings

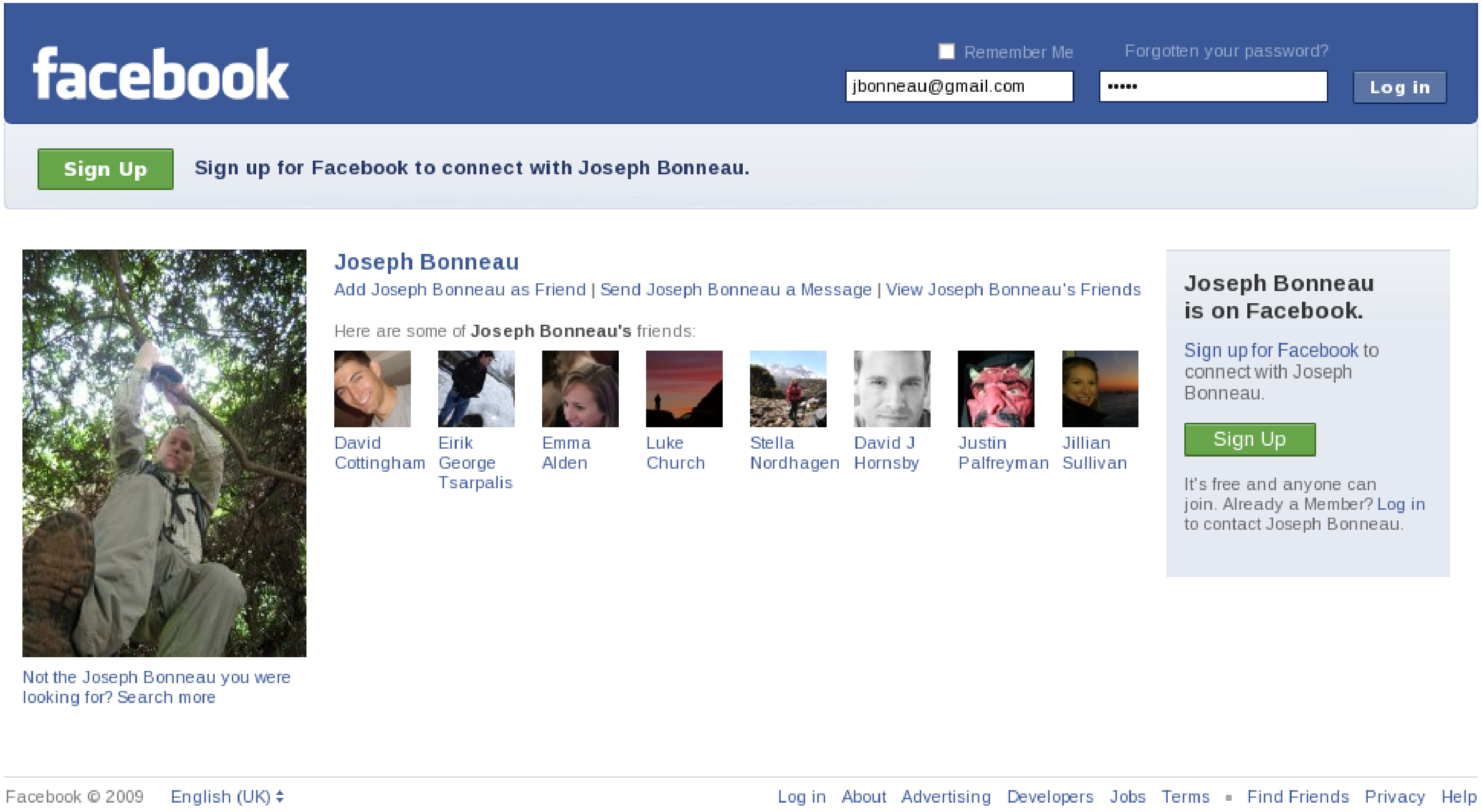
Joseph Bonneau, Jonathan Anderson, Ross Anderson, Frank Stajano

University of Cambridge Computer Laboratory

# Facebook Features & Privacy Backlashes

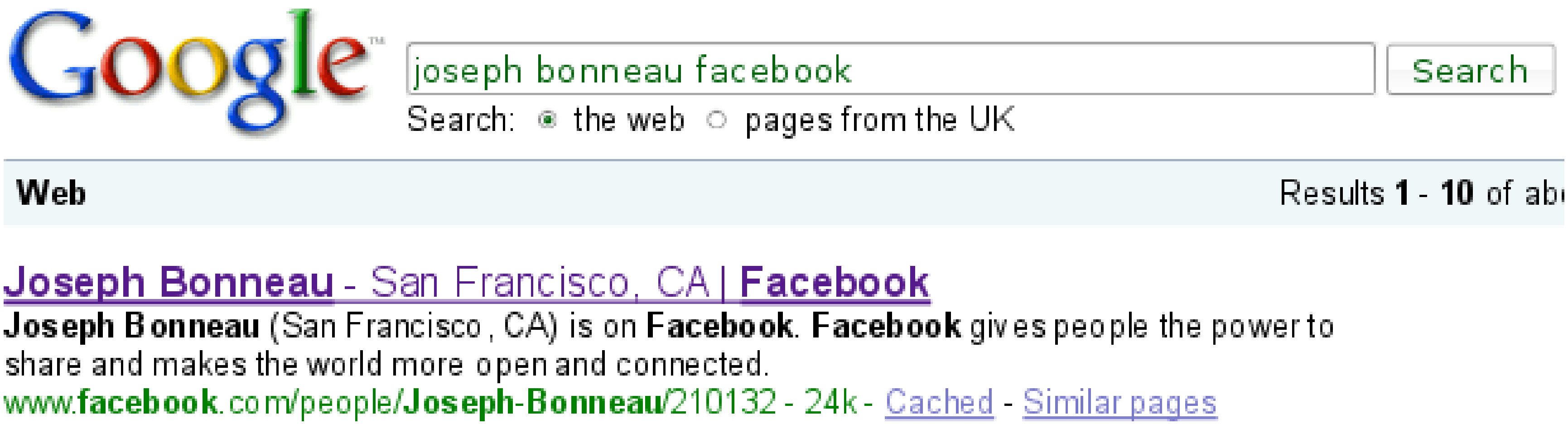
- News Feed (Sep 2006)
- Beacon (Nov 2007)
- “New Facebook” (Sep 2008)
- Terms of Use (Feb 2009)
- New Product Pages (Mar 2009)

# A Quietly Introduced Feature...



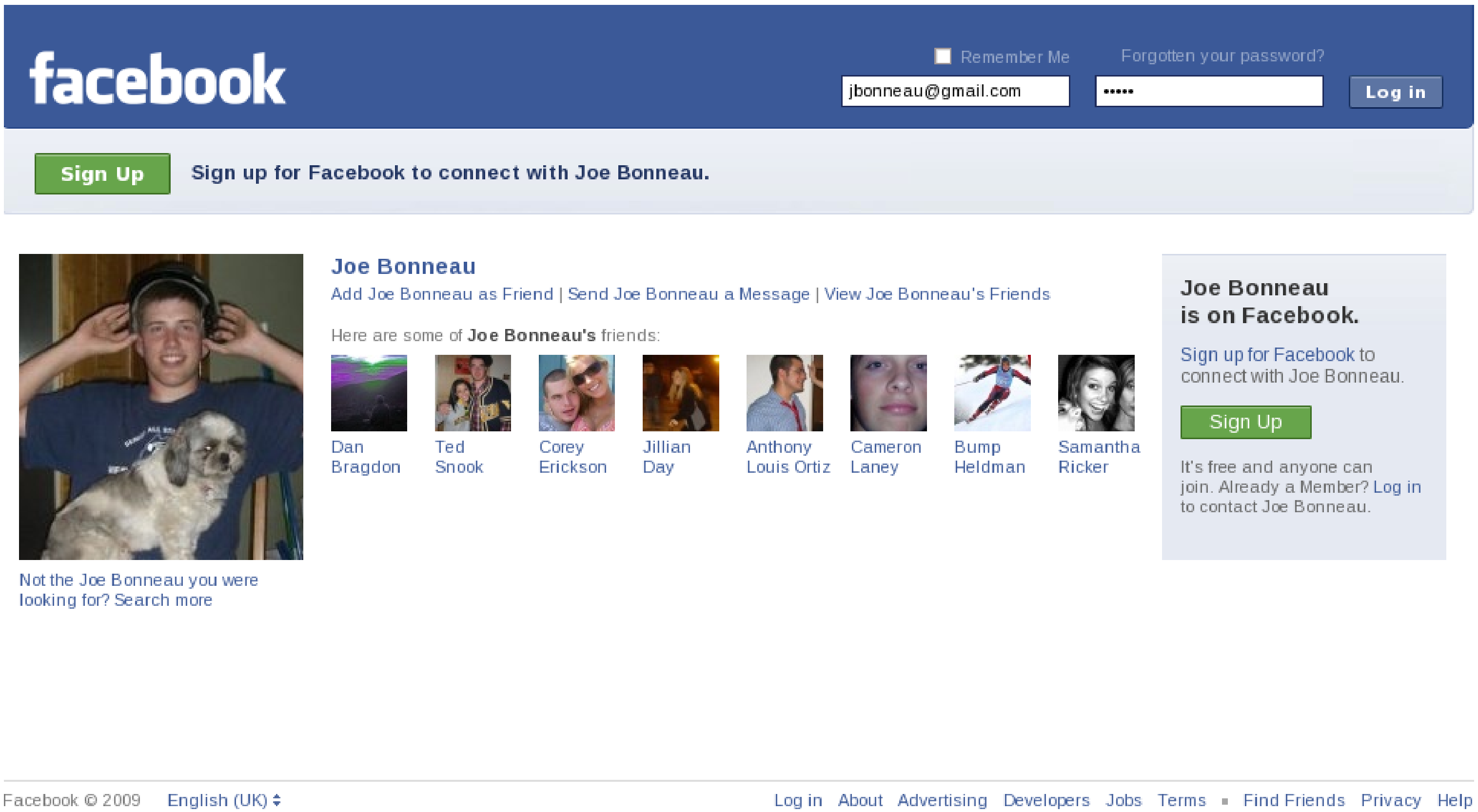

Public Search Listings, Sep 2007

# Public Search Listings




- Unprotected against crawling
- Indexed by search engines
- Opt out—but most users don't know it exists!

# Utility




Entity Resolution

# Utility

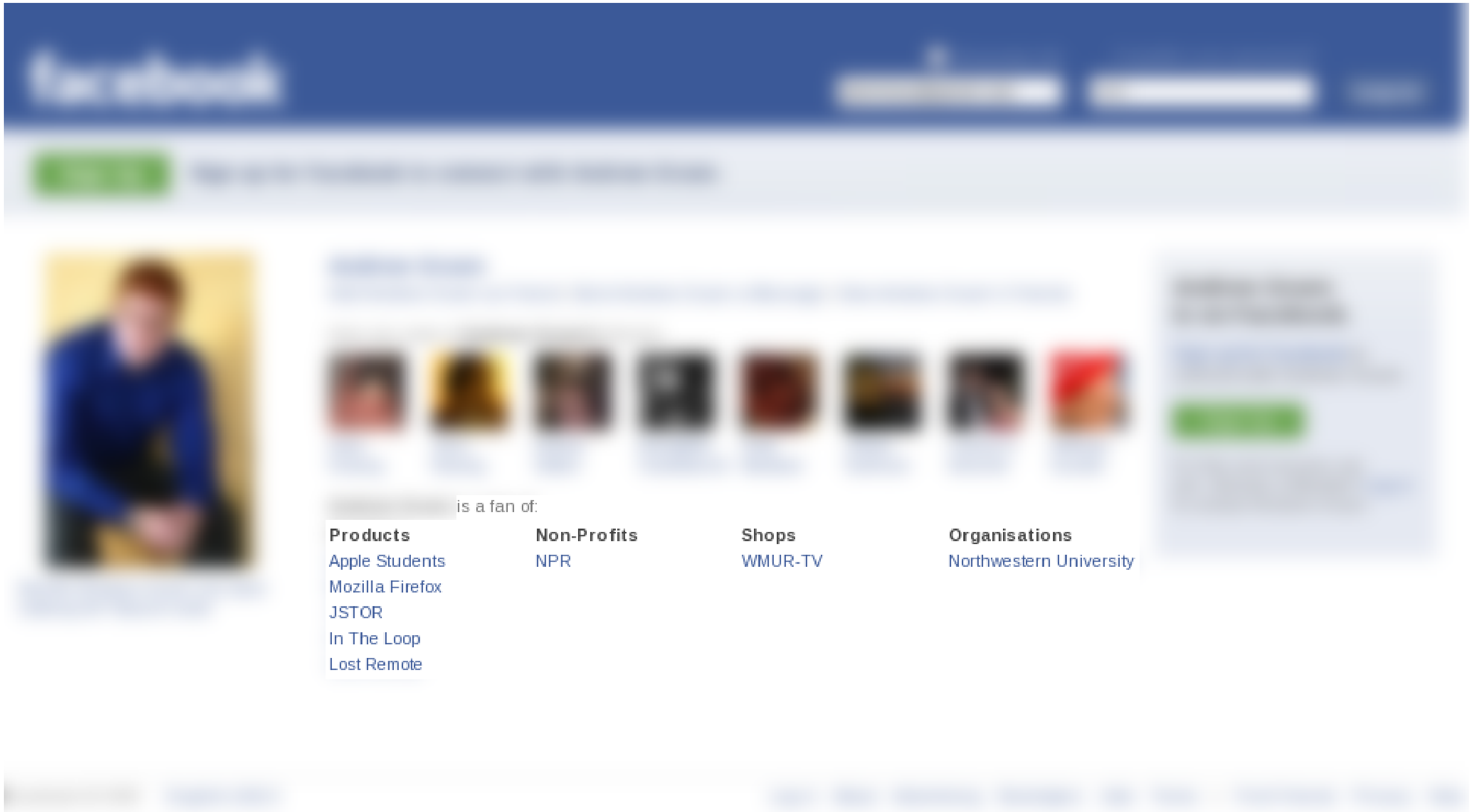

Promotion via Network Effects

# Legal Status

*"Your name, network names, and profile picture thumbnail will be available in search results across the Facebook network and those limited pieces of information may be made available to third party search engines. This is primarily so your friends can find you and send a friend request."*

-Facebook Privacy Policy

# Legal Status




Much More Info Now Included...

# Legal Status

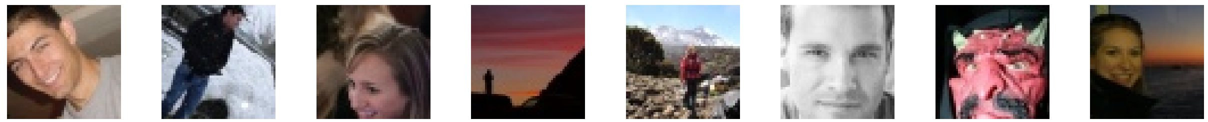

Public Group Pages Recently Added

# Obvious Attack

- Initially returned new friend set on refresh
- Can find all *n* friends in O(*n*·log *n*) queries
  - The Coupon Collector's Problem
  - For 100 Friends, need 65 page refreshes
- As of Jan 2009, friends fixed per IP address

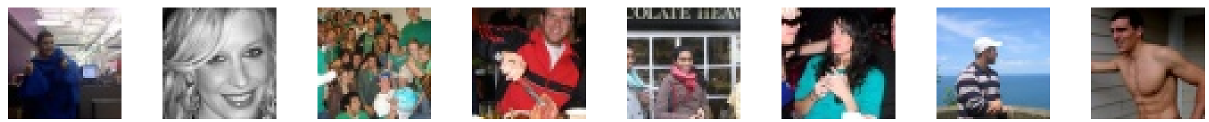
# Fun with Tor

UK



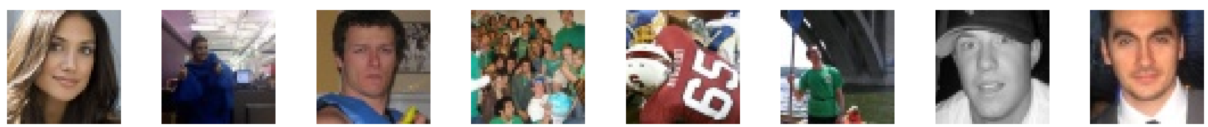
David Cottingham | Eirik George Tsarpalis | Emma Alden | Luke Church | Stella Nordhagen | David J Hornsby | Justin Palfreyman | Jillian Sullivan

Germany



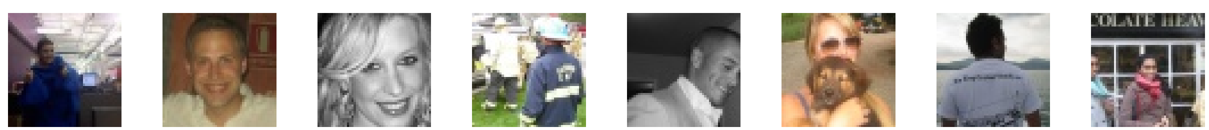
Shoshana Freisinger | Lauren Duffey | Conor Loftus-Sweetland | Will Cordingley | Srilakshmi Raj | Sarita Kristina Sylvester | Brian Brown | Gary Champagne

USA



Melanie Kannokada | Shoshana Freisinger | Russ Heddleston | Conor Loftus-Sweetland | Gustav Rydstedt | Seth Ort | Cameron Lochte | Ben Skolnik

Australia



Shoshana Freisinger | Federico Baradello | Lauren Duffey | Adrian Boscolo-Hightower | Justin David Carl | Katie Gunderson | Ankit Garg | Srilakshmi Raj

# Attack Scenario

- Spider all public listings
  - Our experiments crawled 250 k users daily
  - Implies ~800 CPU-days to recover all users
- Compute functions on sampled graph

# Abstraction

- Take a graph $G = <V,E>$
- Randomly select $k$ out-edges from each node
- Result is a sampled graph $G_k = <V,E_k>$
- Try to approximate $f(G) \approx f_{\text{approx}}(G_k)$

# Approximable Functions

- Node Degree
- Dominating Set
- Betweenness Centrality
- Path Length
- Community Structure

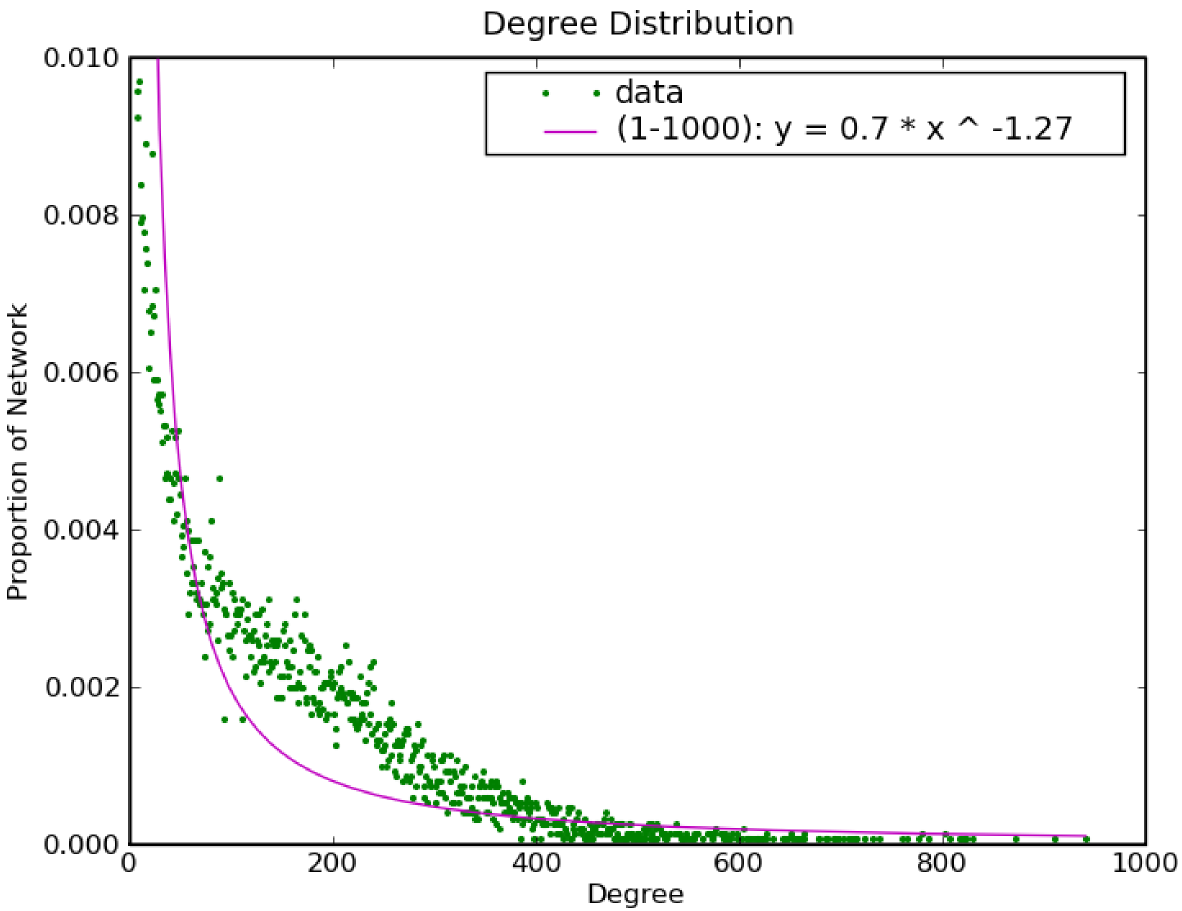
# Experimental Data

- Crawled networks for Stanford, Harvard universities

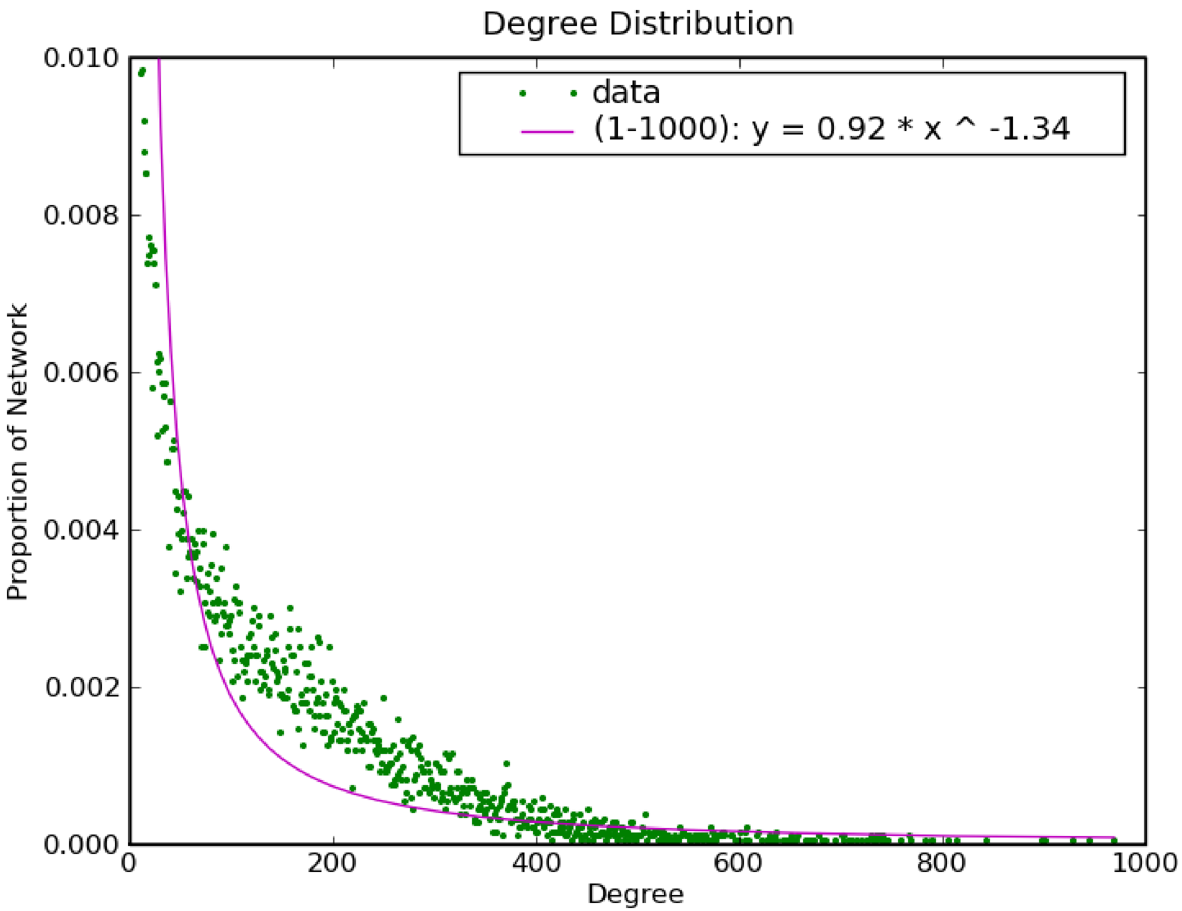
- Representative sub-networks

| | # Users | Mean *d* | Median *d* |
|---|---|---|---|
| Stanford | 15043 | 125 | 90 |
| Harvard | 18273 | 116 | 76 |

# Stanford Histogram

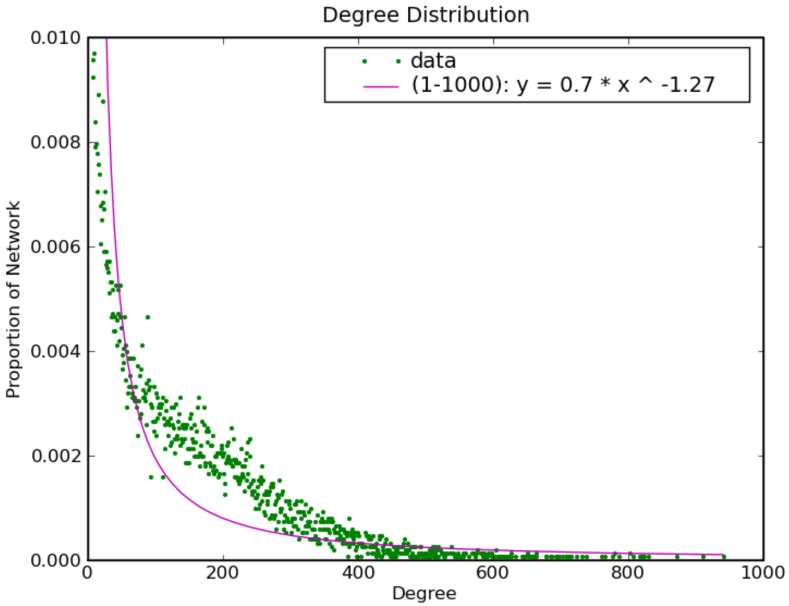


# Harvard Histogram

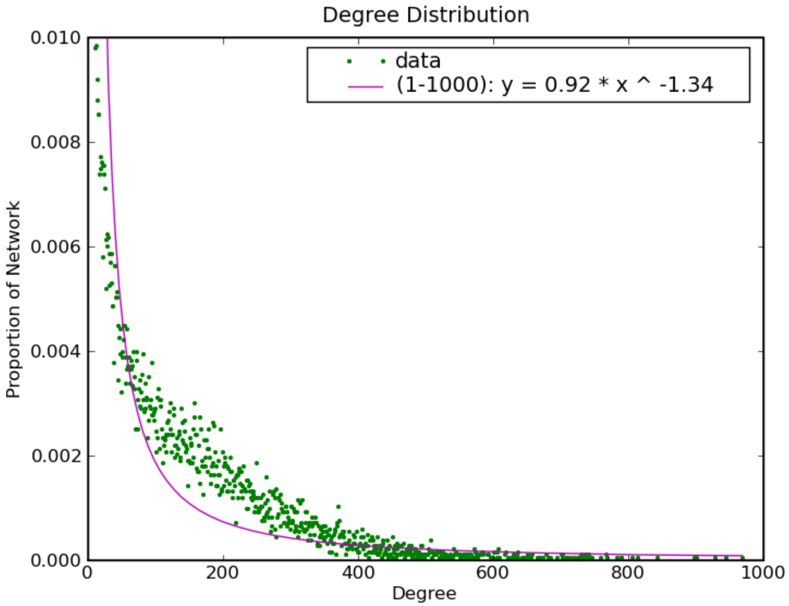


# Comparison

Stanford

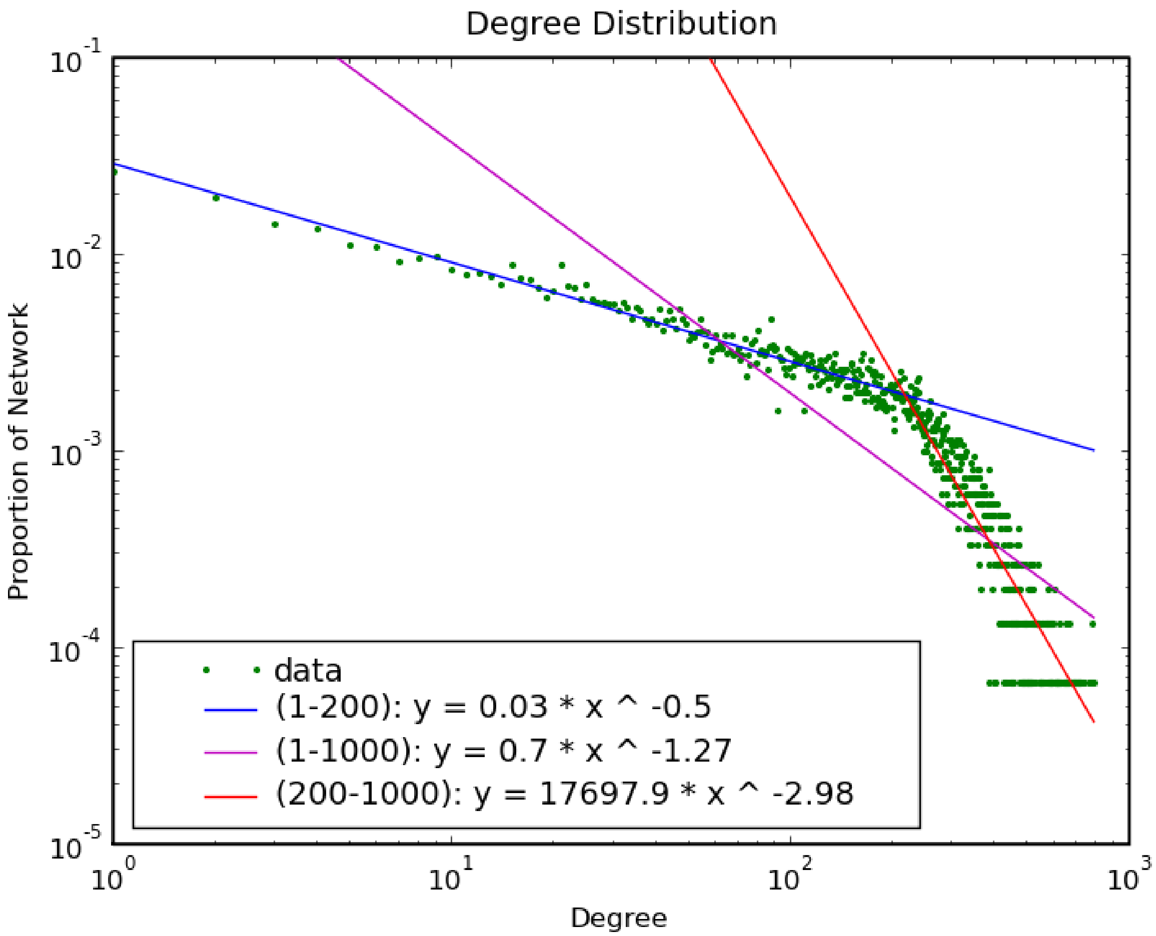



Harvard

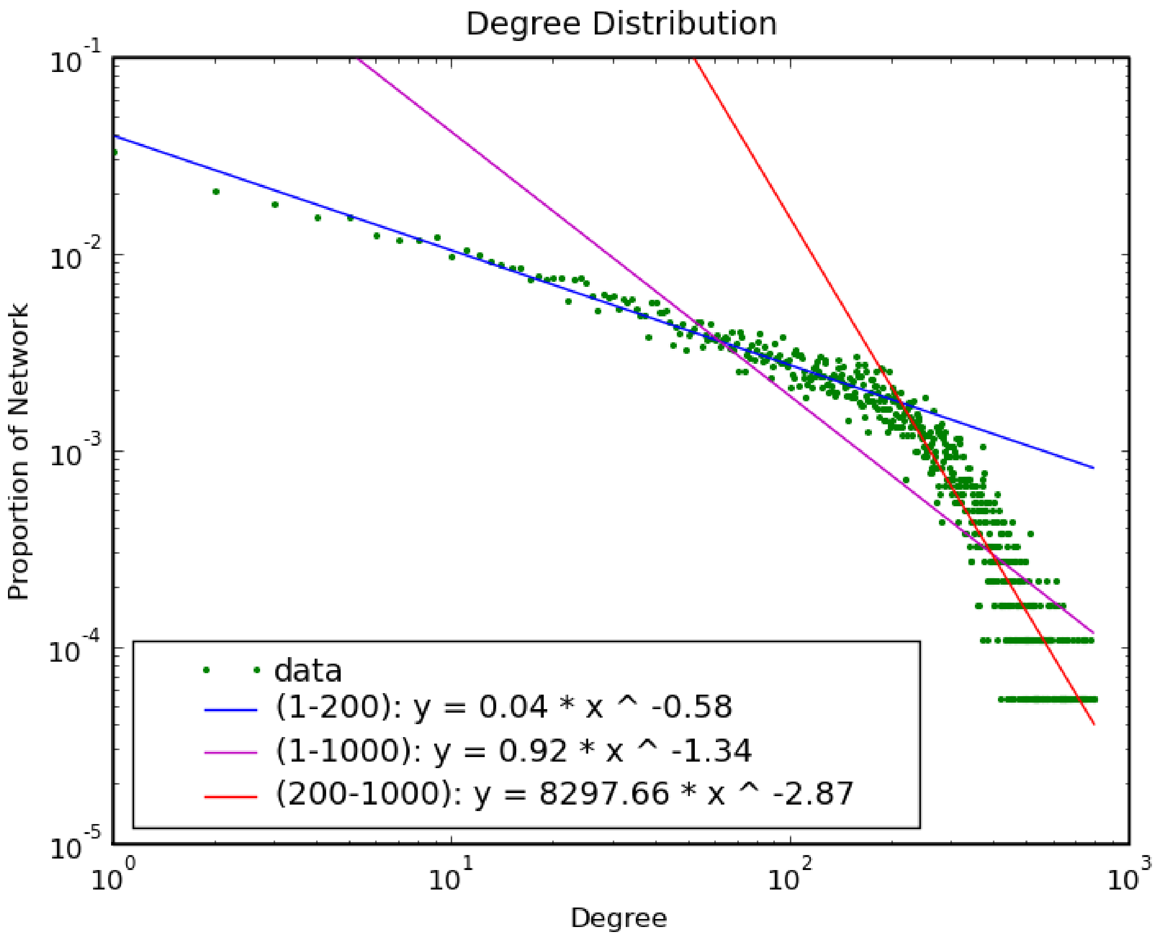



Networks have very similar structure

# Stanford Log-Log plot



# Harvard Log-Log plot



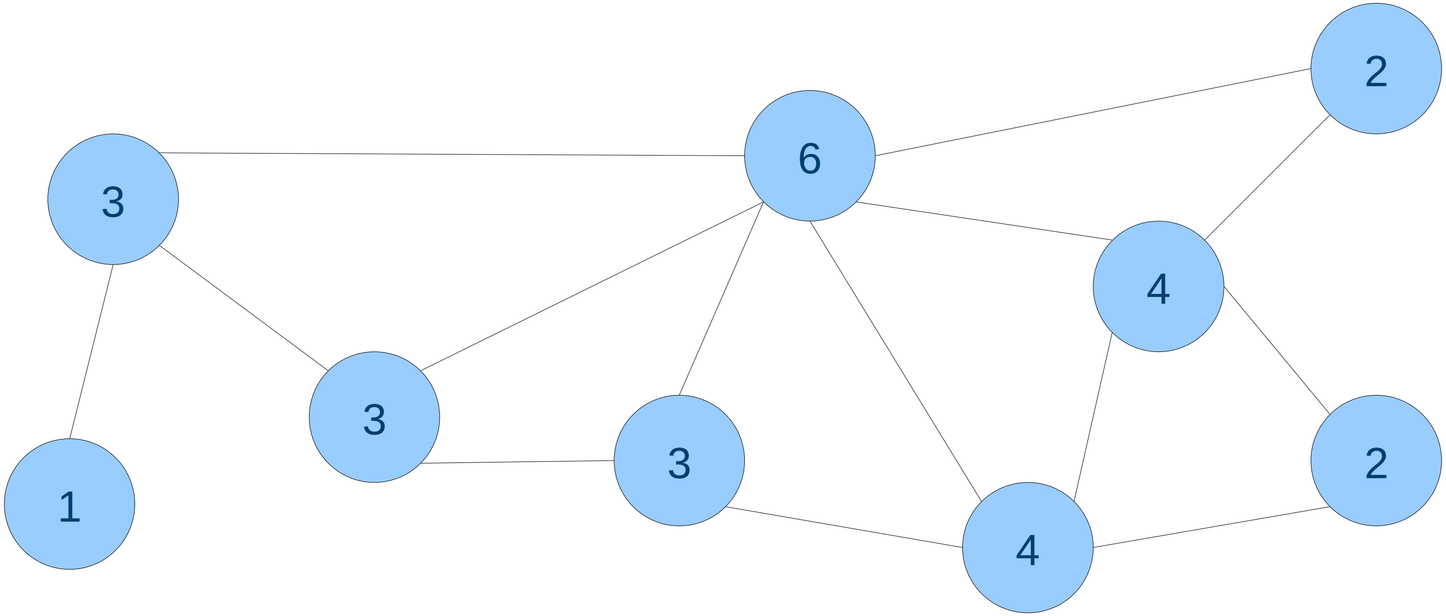
# Back To Our Abstraction

- Take a graph $G = <V,E>$
- Randomly select $k$ out-edges from each node
- Result is a sampled graph $G_k = <V,E_k>$
- Try to approximate $f(G) \approx f_{\text{approx}}(G_k)$

# Estimating Degrees

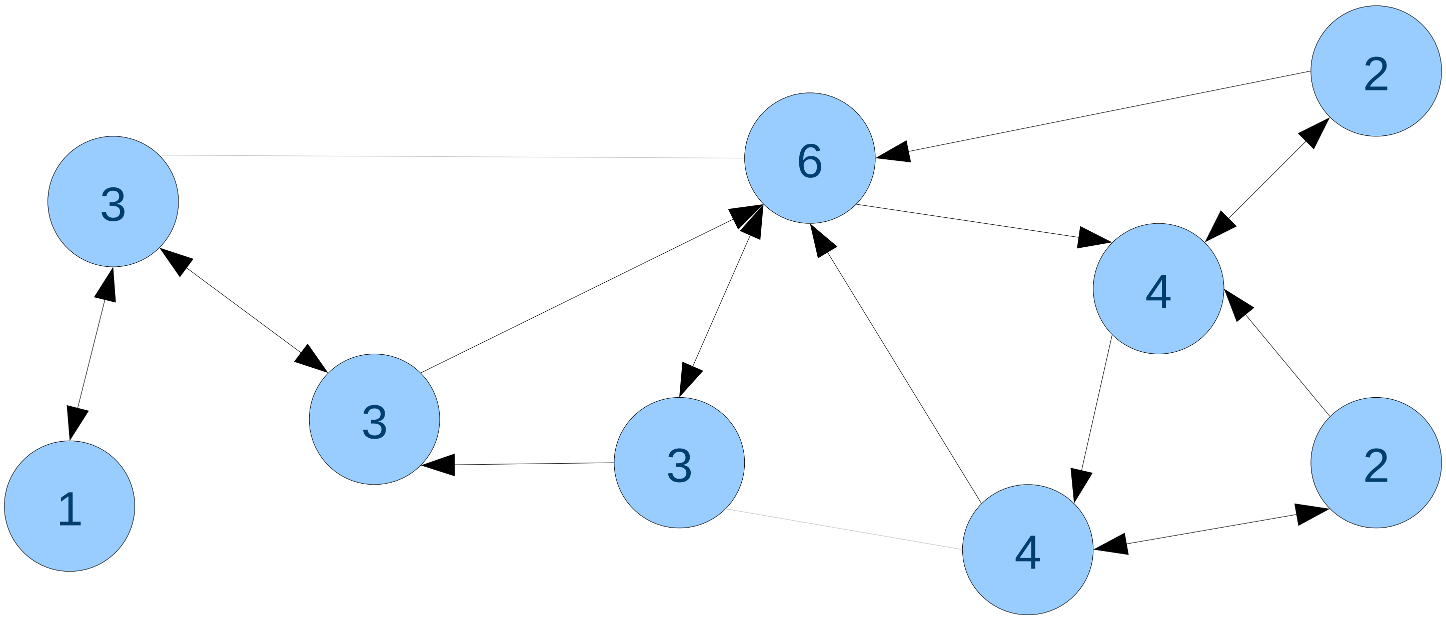
- Convert sampled graph into a directed graph
  - Edges originate at the node where they were seen
- Learn exact degree for nodes with degree < *k*
  - Less than k out-edges
- Get random sample for nodes with degree ≥ *k*
  - Many have more than *k* in-edges

# Estimating Degrees



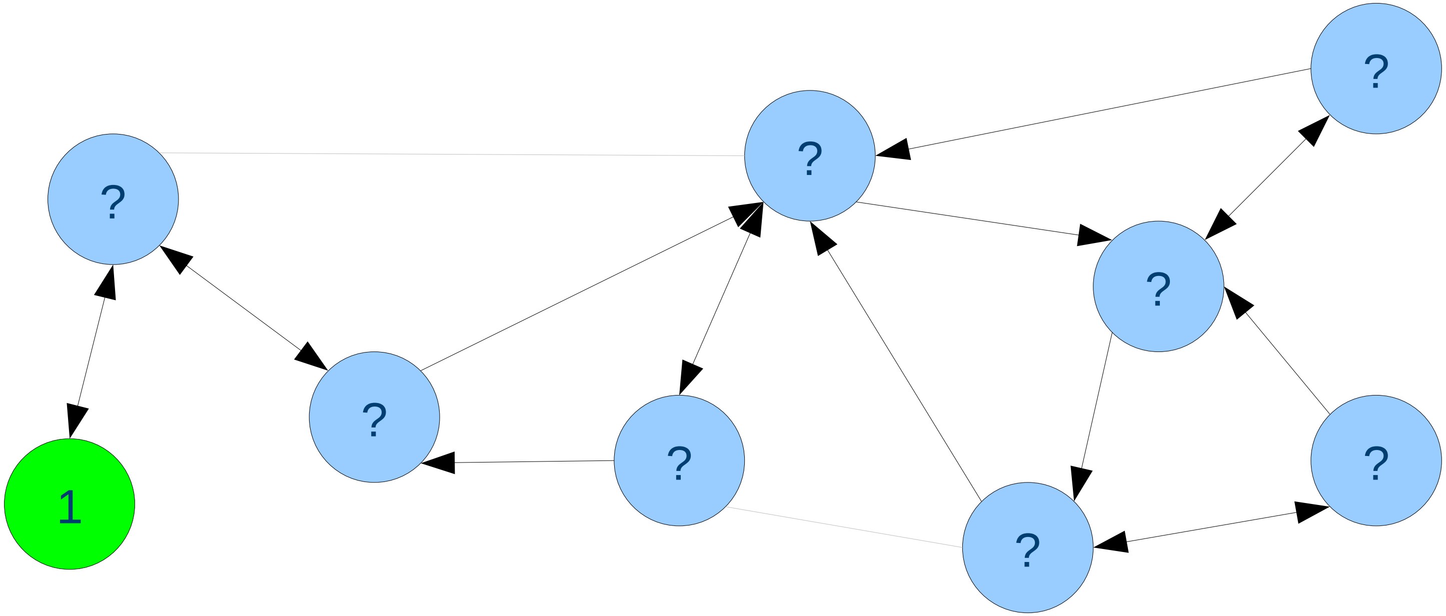

Average Degree: 3.5

# Estimating Degrees



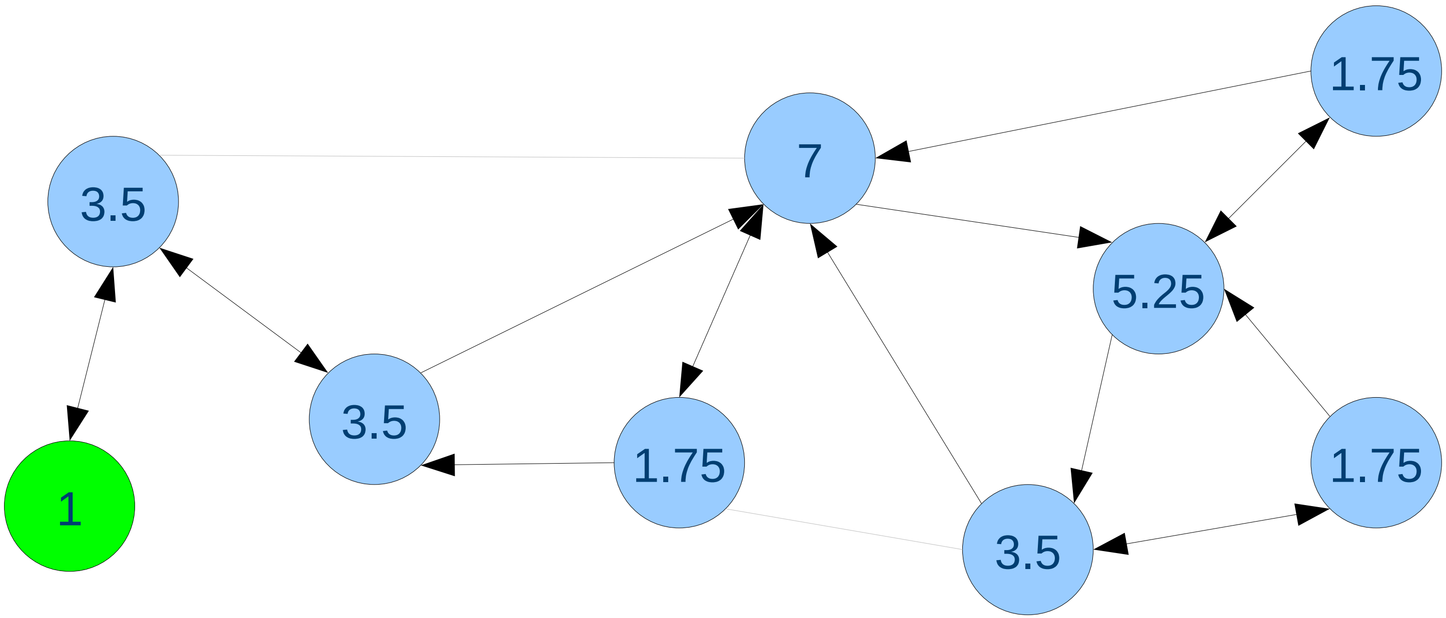

Sampled with *k*=2

# Estimating Degrees



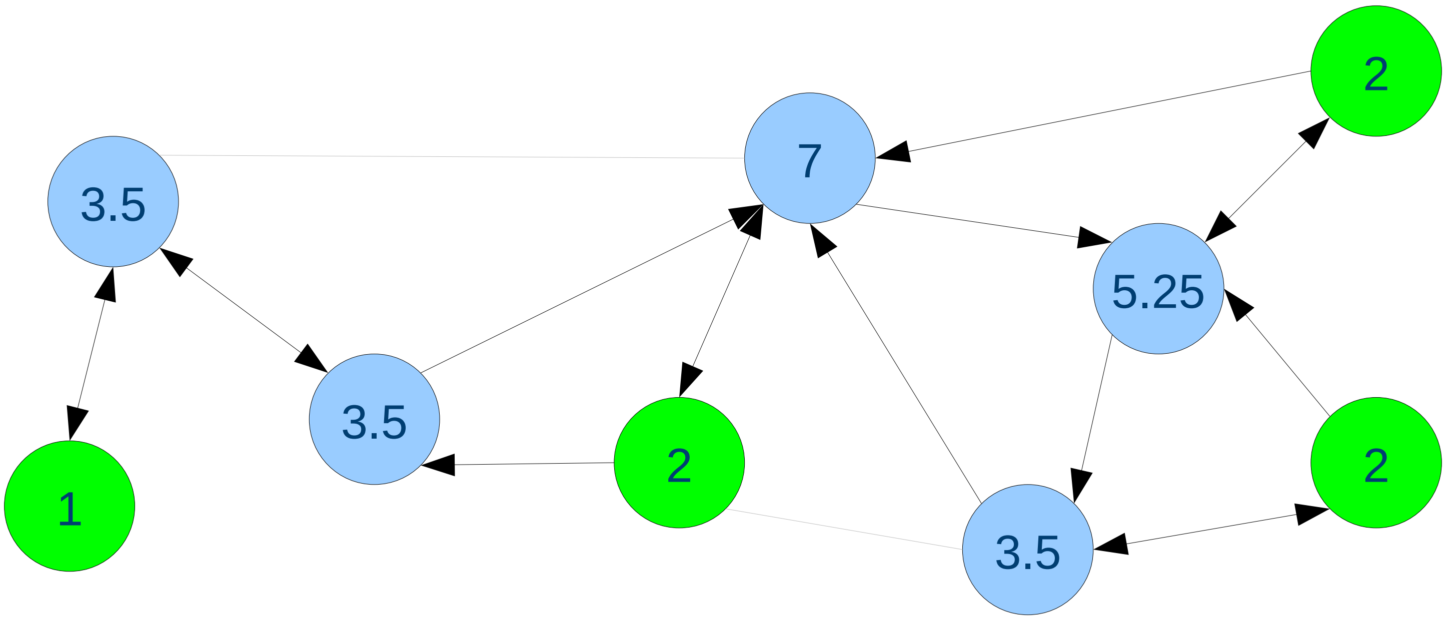

Degree known exactly for one node

# Estimating Degrees



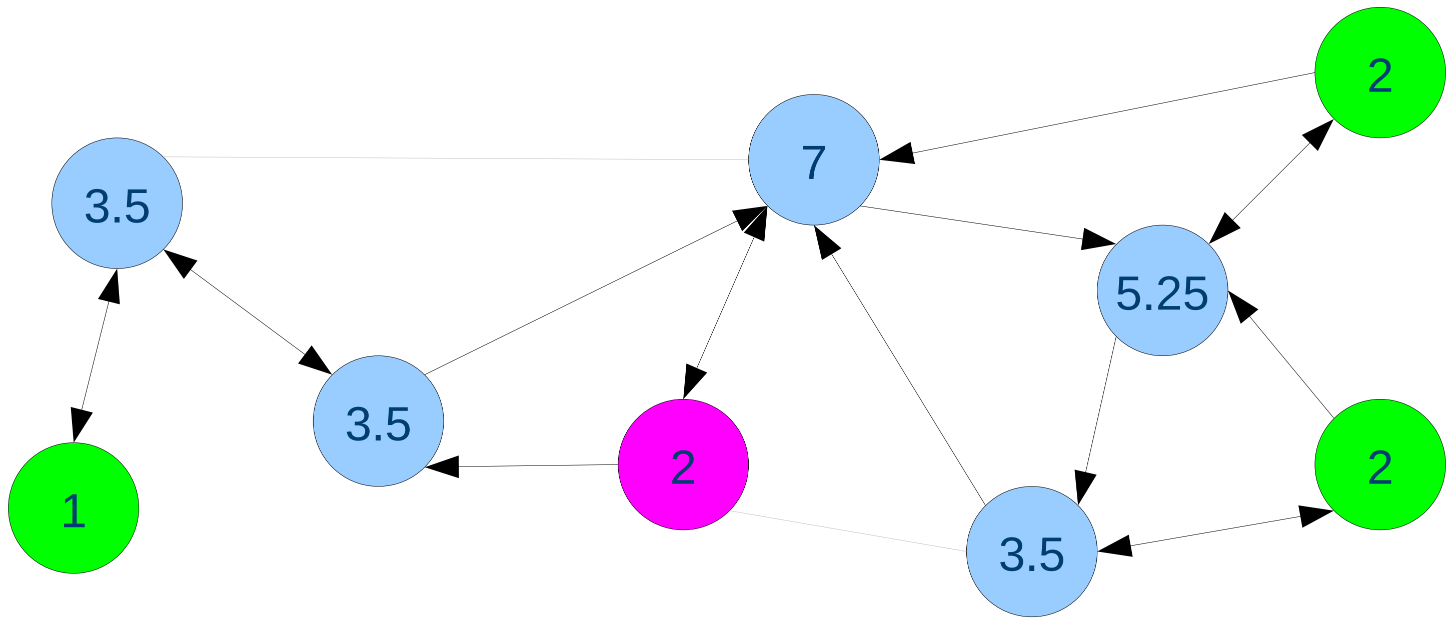

Naïve approach: Multiply in-degree by average degree / *k*

# Estimating Degrees



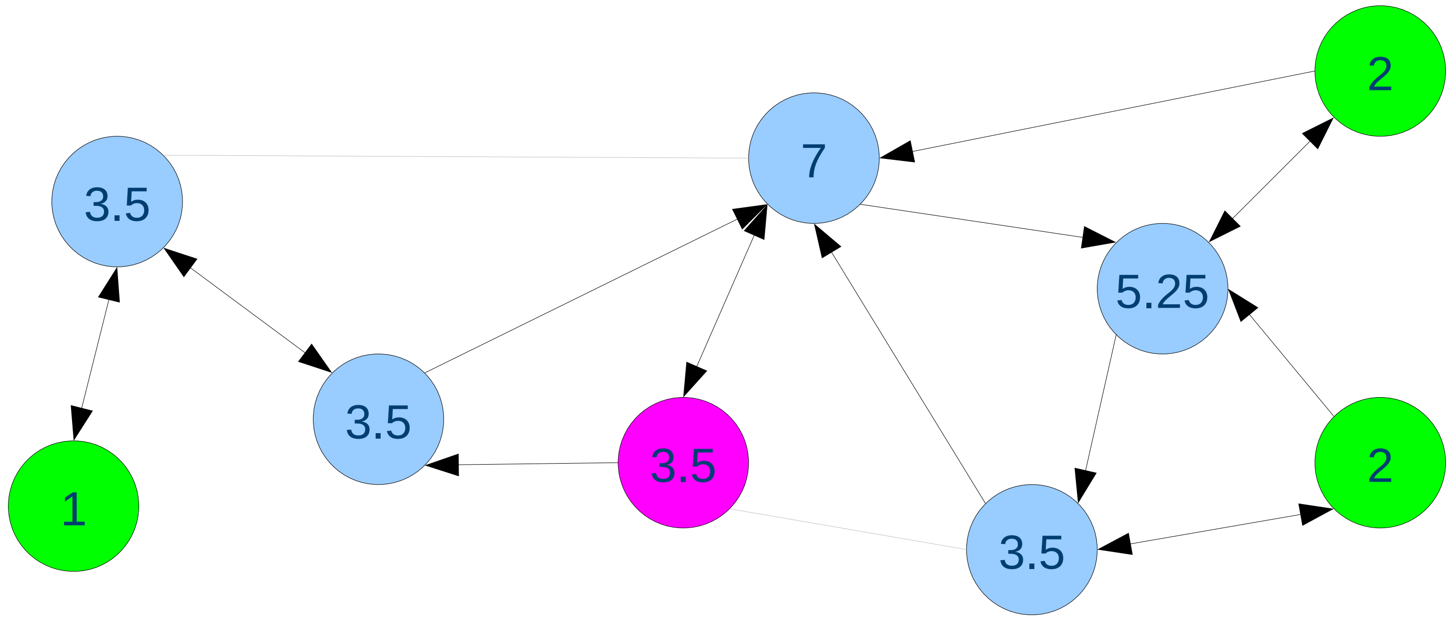

Raise estimates which are less than *k*

# Estimating Degrees



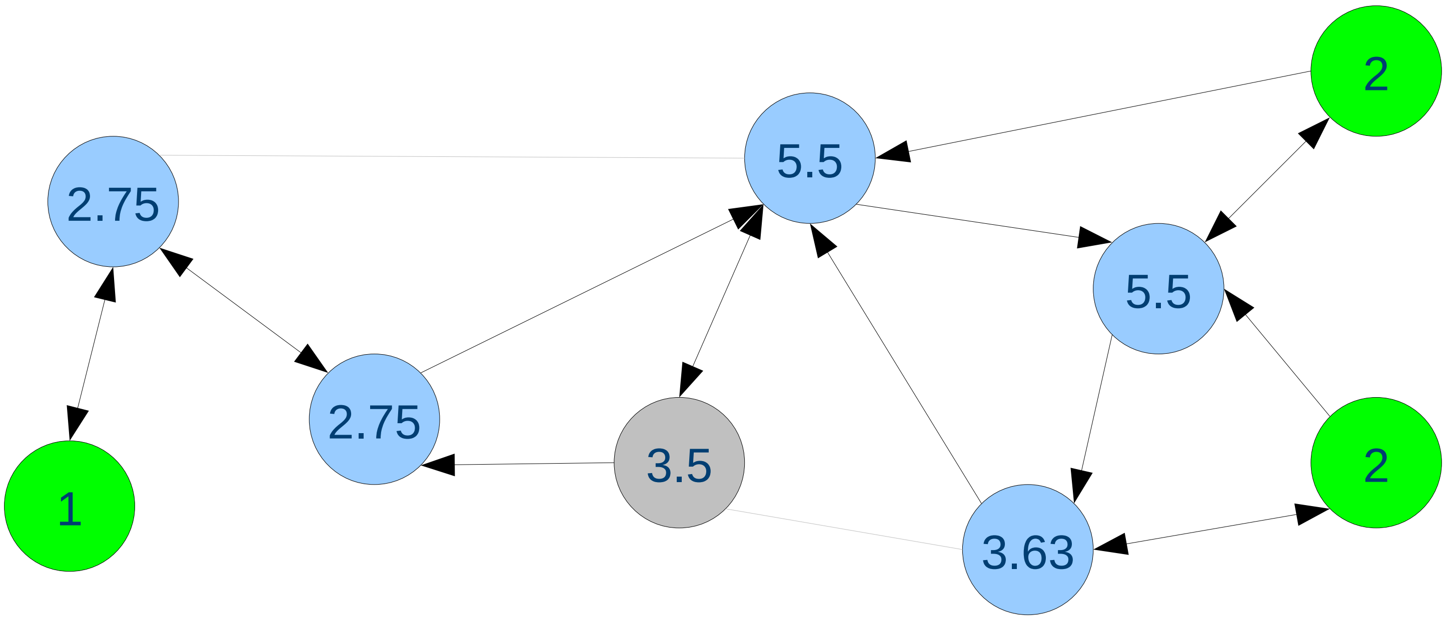

Nodes with high-degree neighbors underestimated

# Estimating Degrees



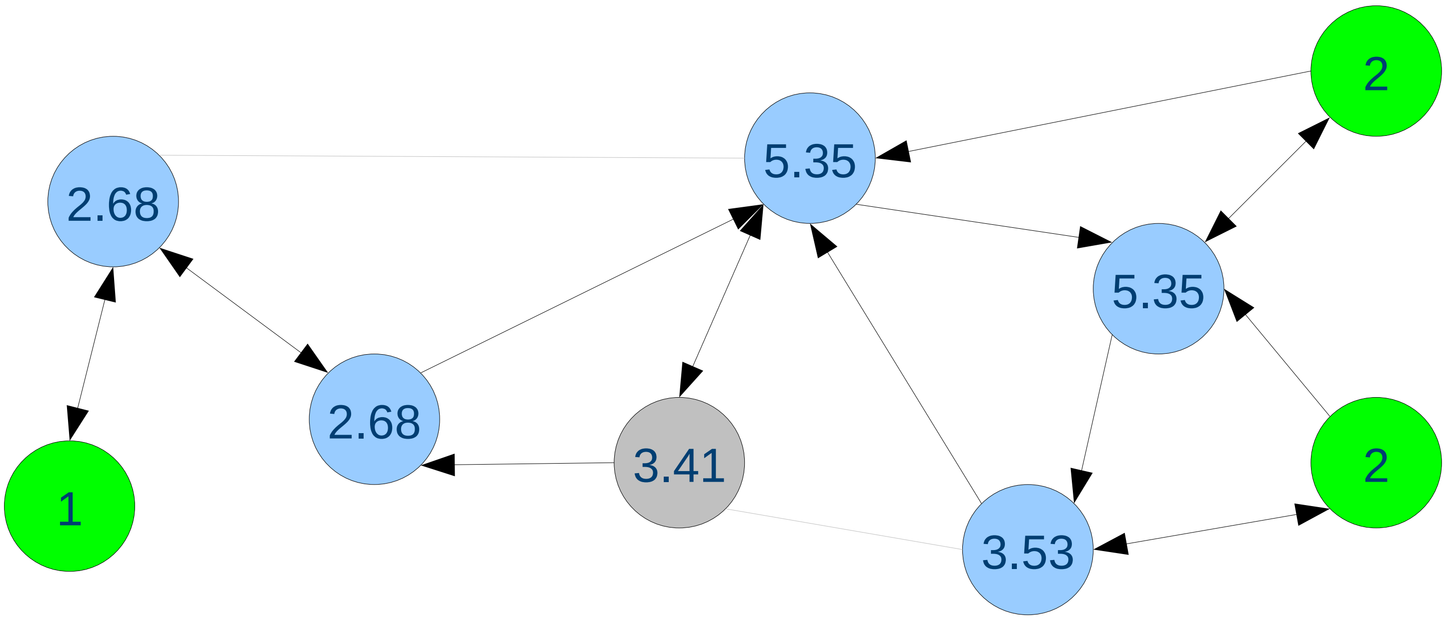

Iteratively scale by current estimate / k in each step

# Estimating Degrees



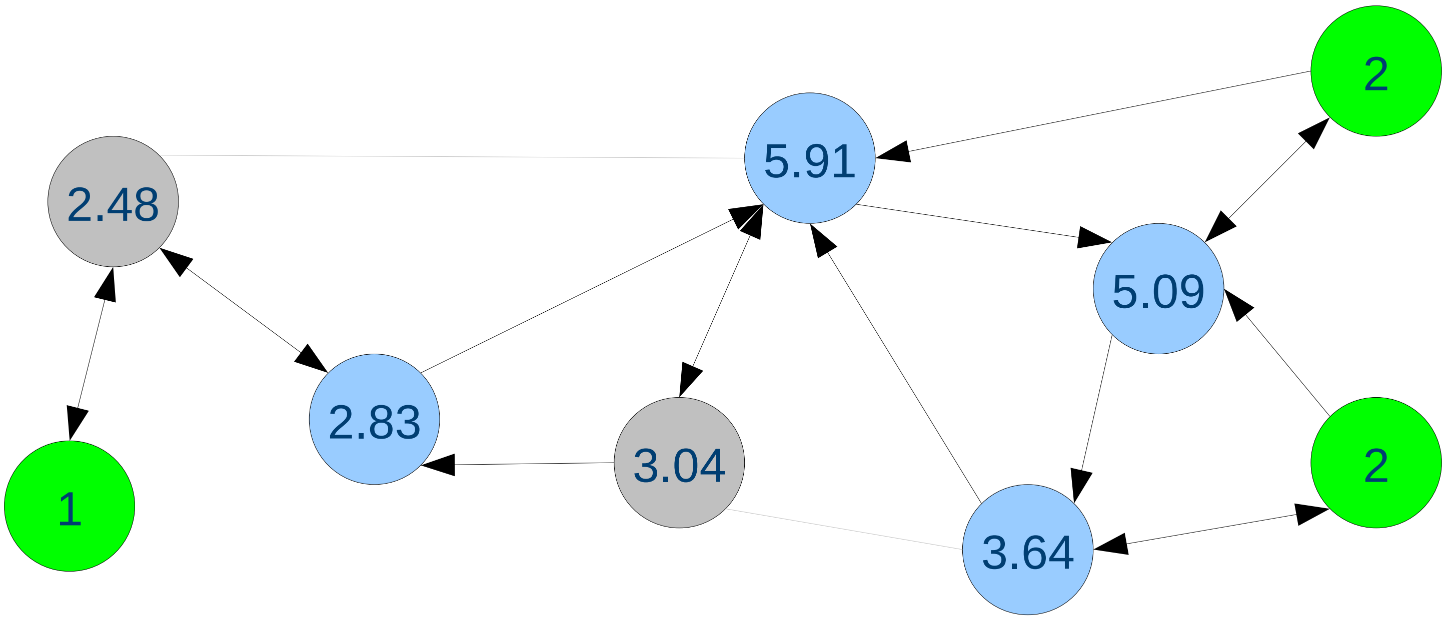

After 1 iteration

# Estimating Degrees




Normalise to estimated total degree

# Estimating Degrees

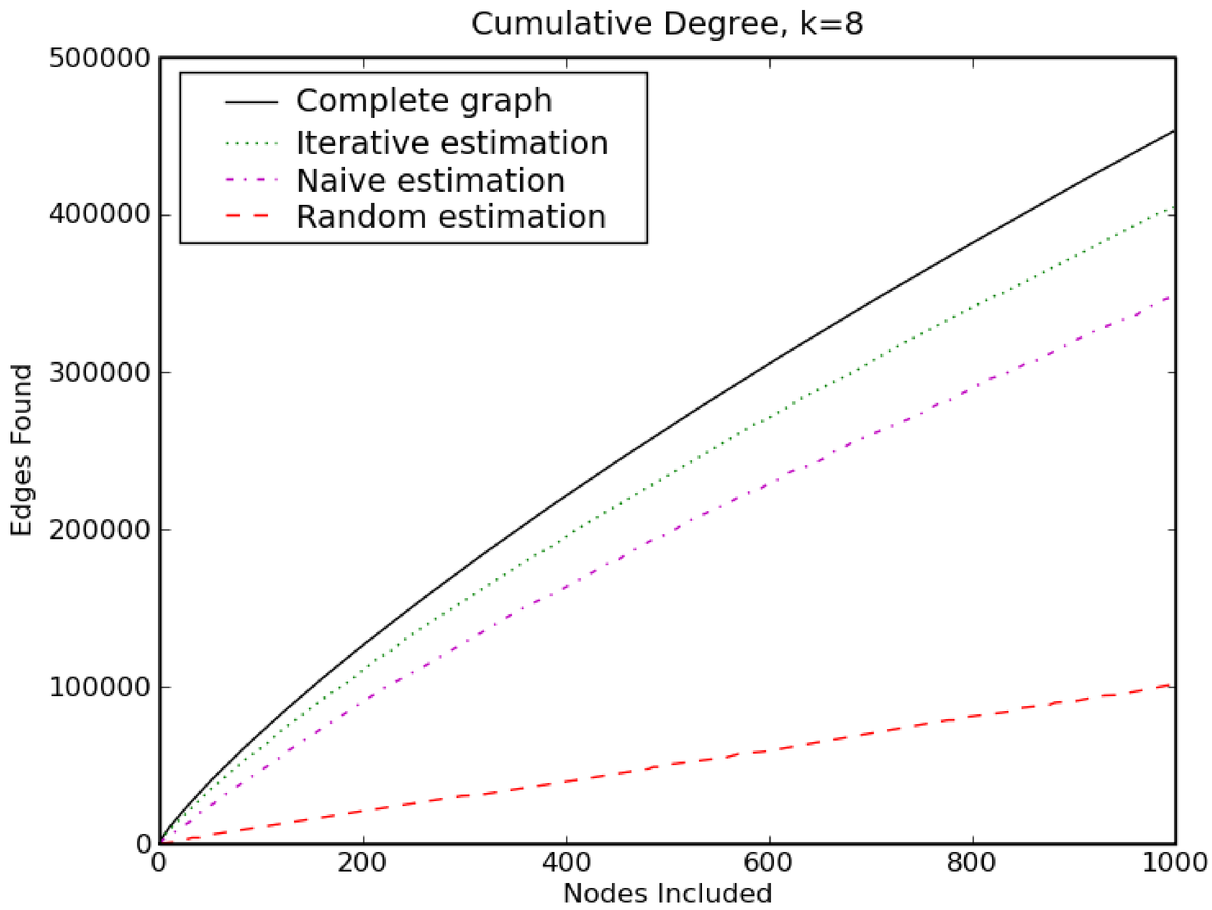



Convergence after n > 10 iterations

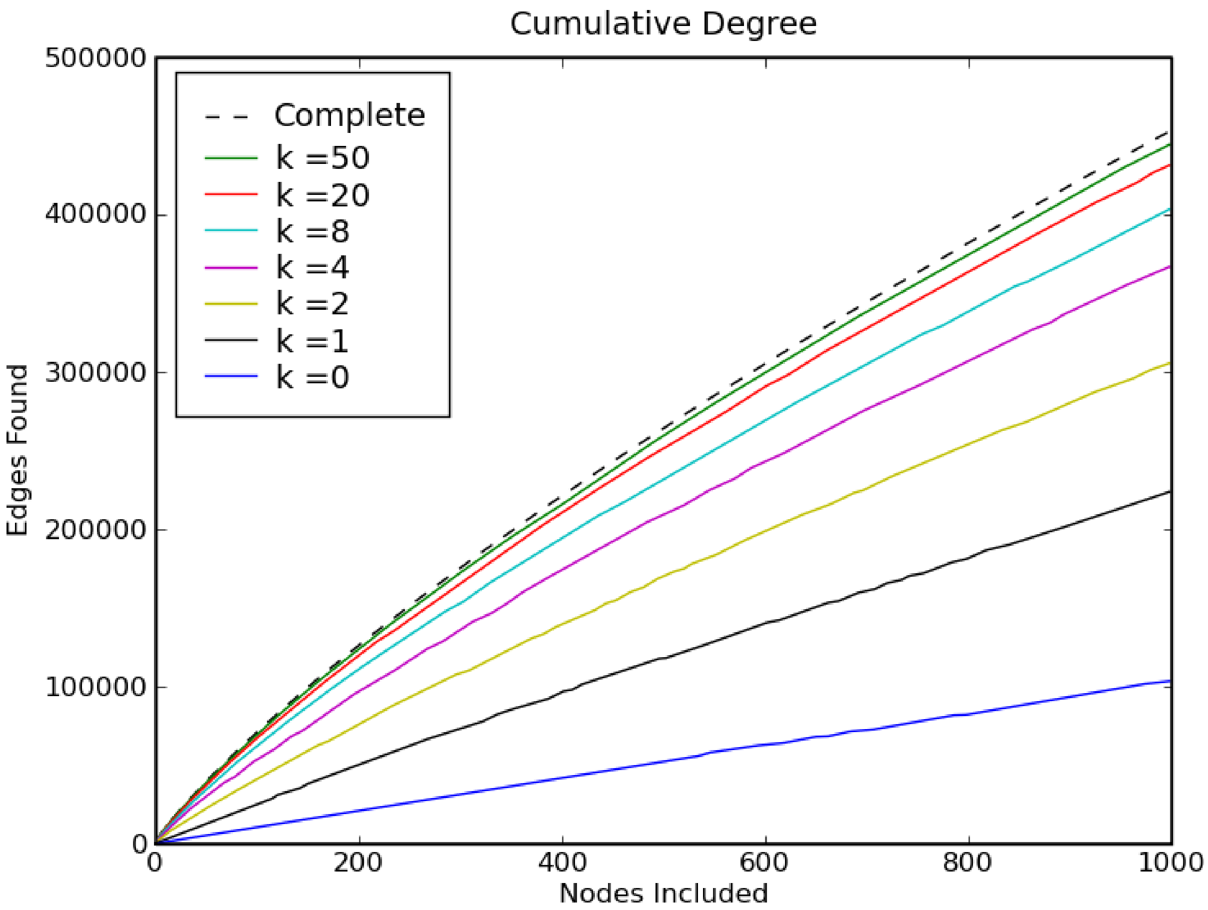
# Estimating Degrees

- Converges fast, typically after 10 iterations
- Absolute error is high—38% average
  - Reduced to 23% for nodes with $d \geq 50$
- Still accurately can pick high degree nodes

# Aggregate of *x* highest-degree nodes



# Comparison of sampling parameters



# Dominating Sets

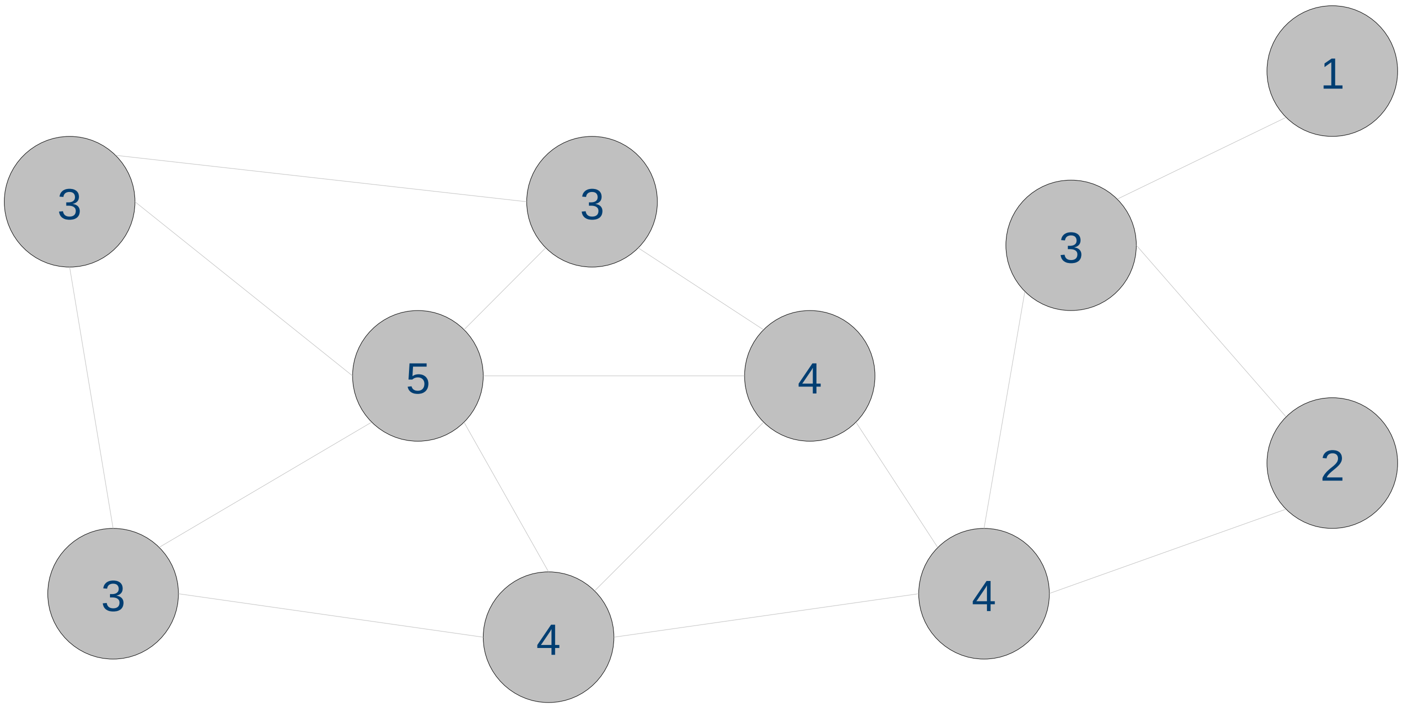
- Set of Nodes $D \subseteq V$ such that

  $D \cup \text{Neighbours}(D) = V$

- Set allows viewing the entire network

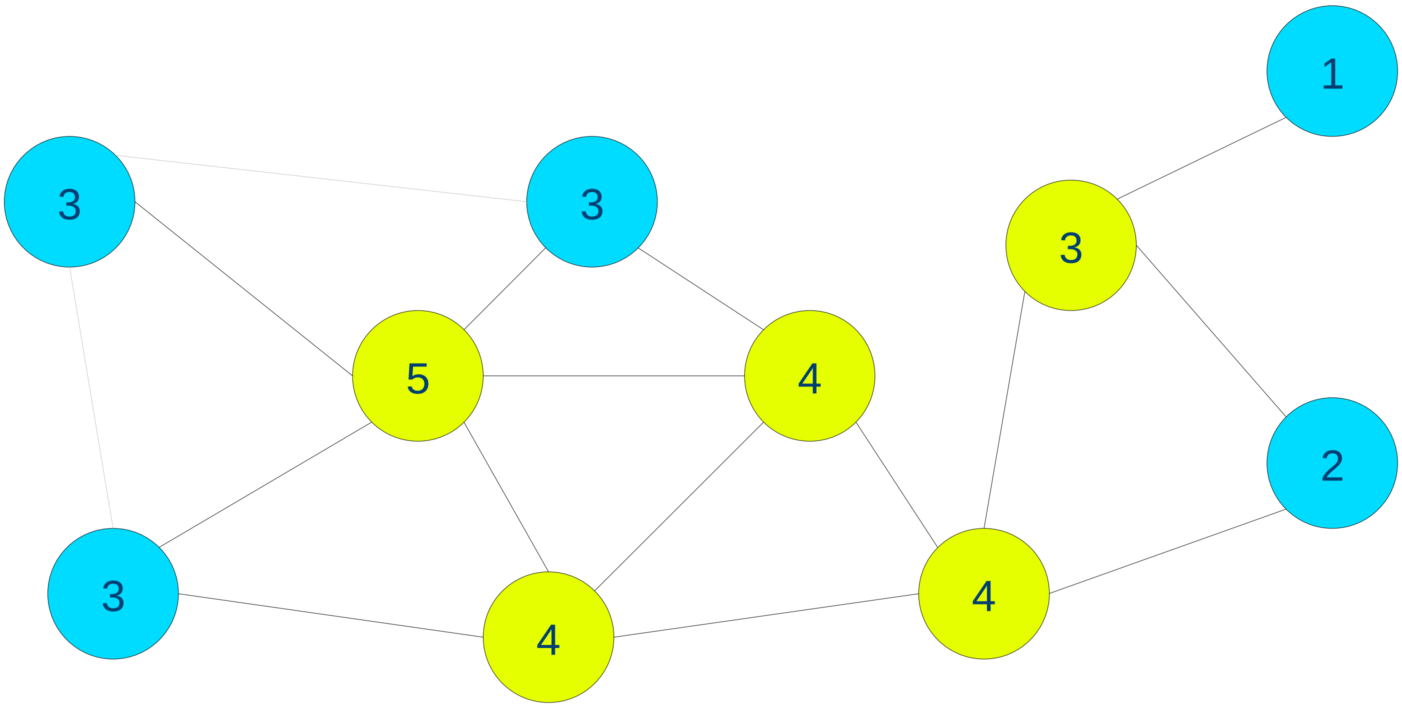
- Also useful for marketing, trend-setting

# Dominating Sets



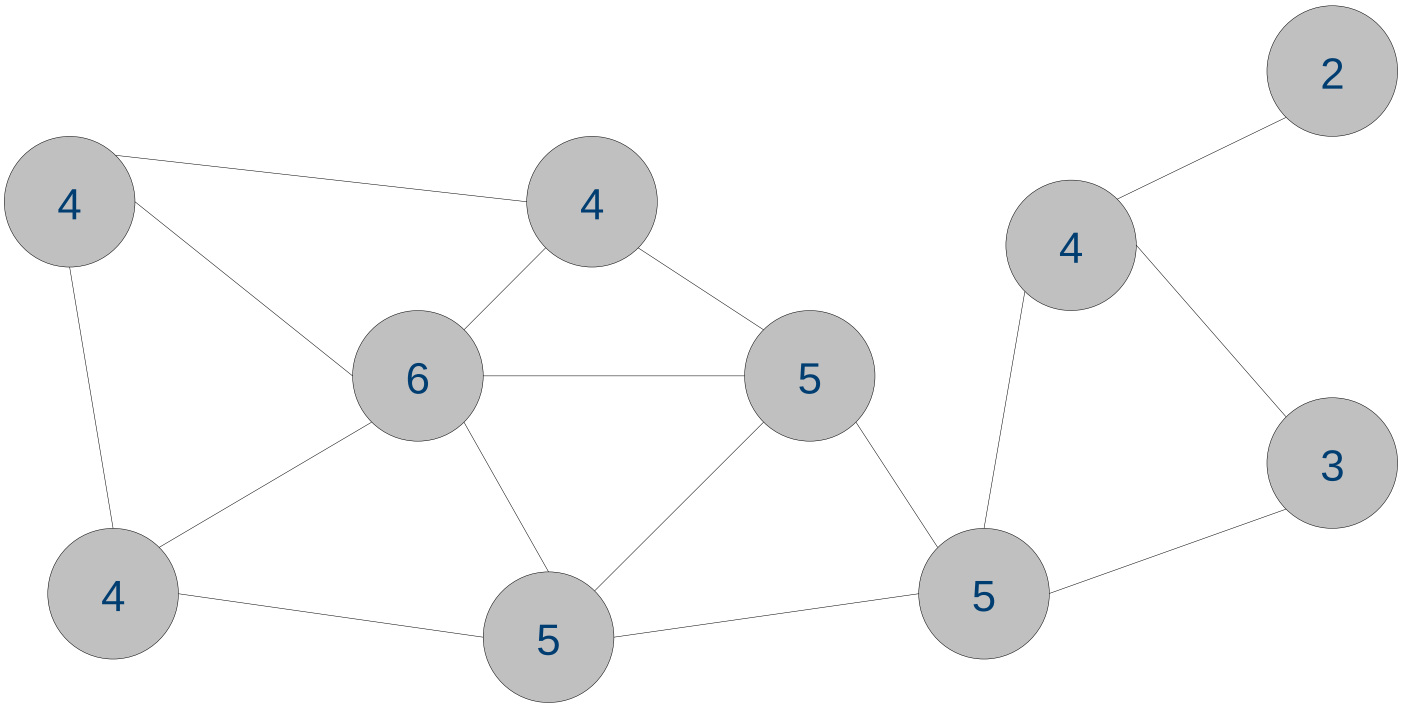

Trivial Algorithm: Select High-Degree Nodes in Order

# Dominating Sets



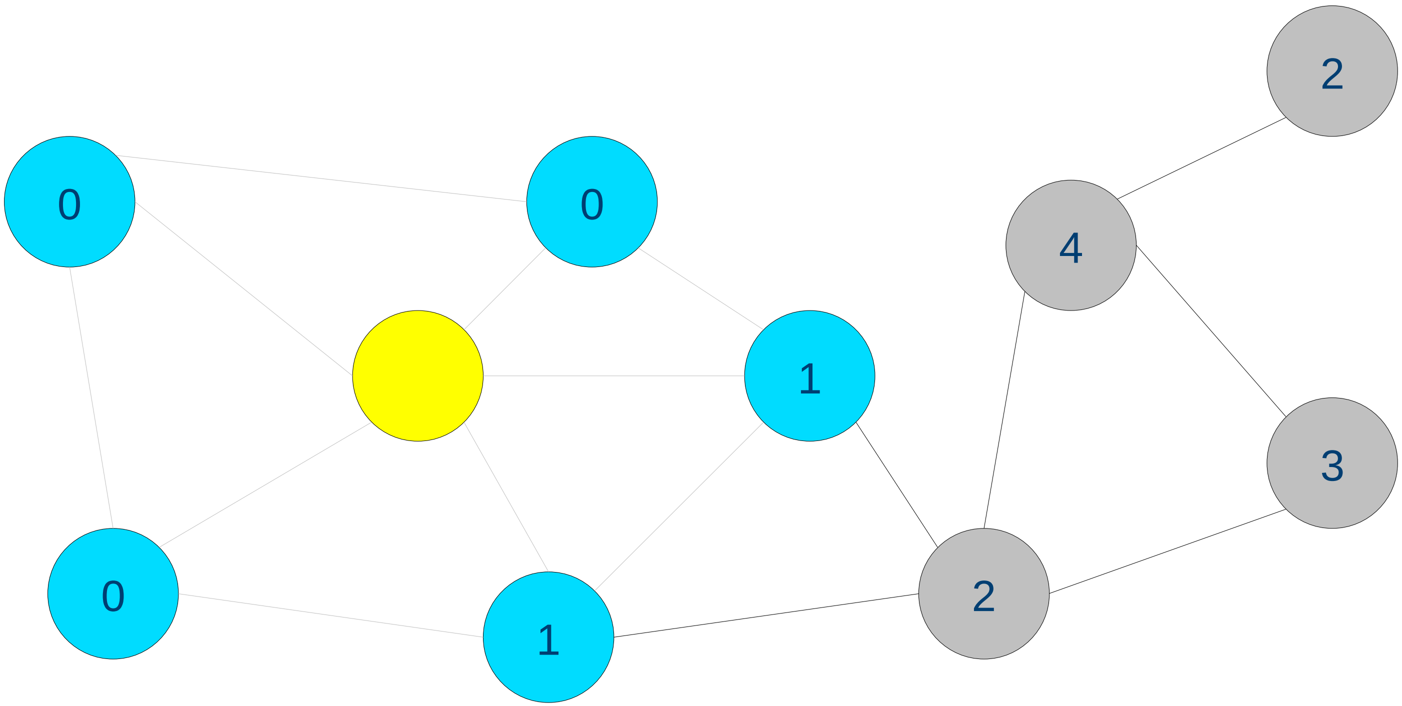

In fact, finding minimal dominating set is NP-complete

# Dominating Sets



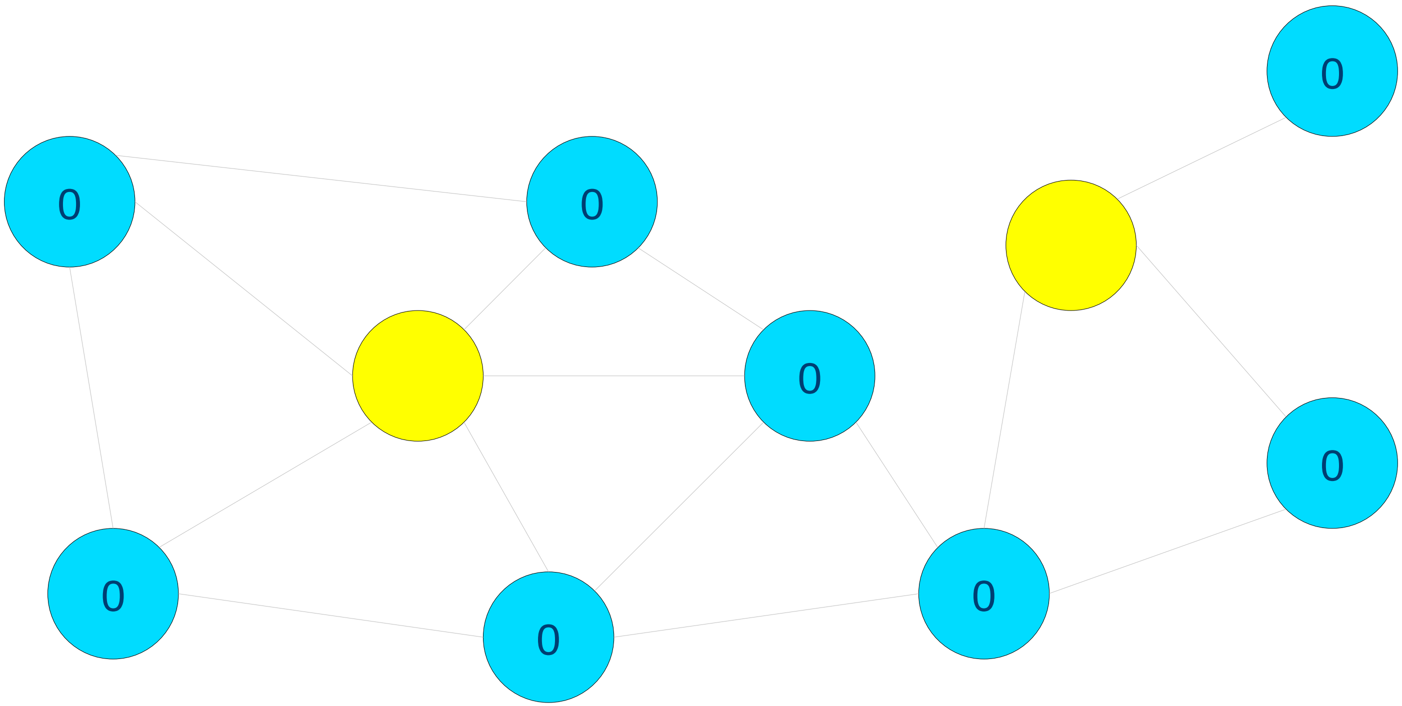

Greedy Algorithm: select for maximal coverage

# Dominating Sets



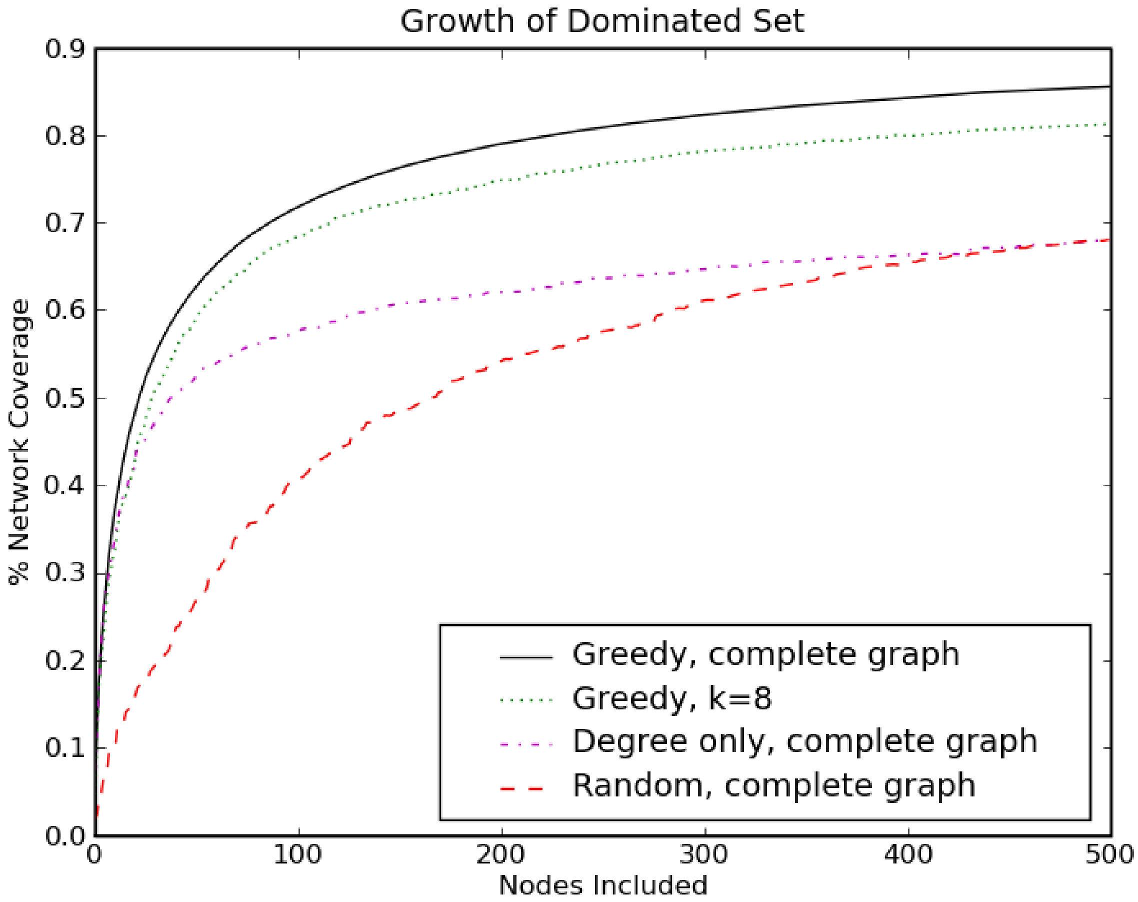

Greedy Algorithm: select for maximal coverage

# Dominating Sets

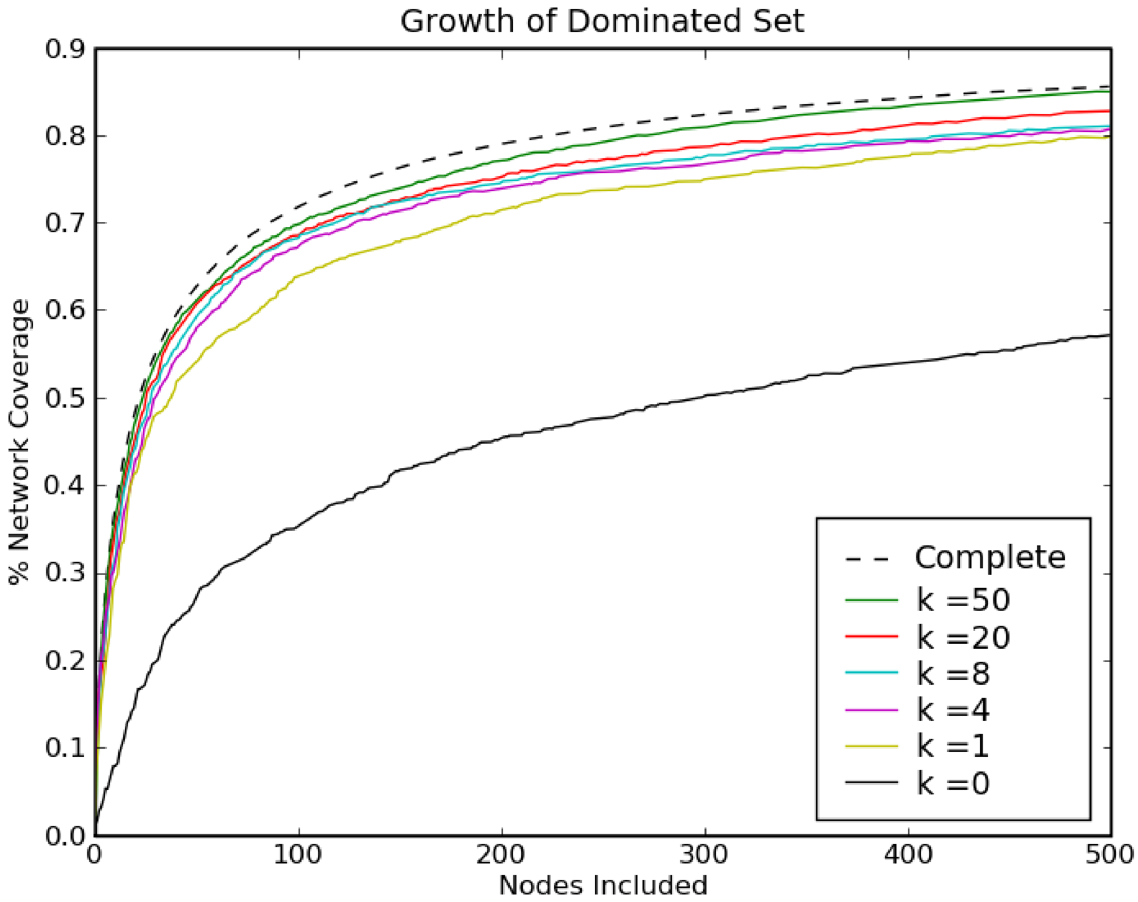



Shown to perform adequately in practice

# Works Well on Sampled Graph



# Insensitive to Sampling Parameter!



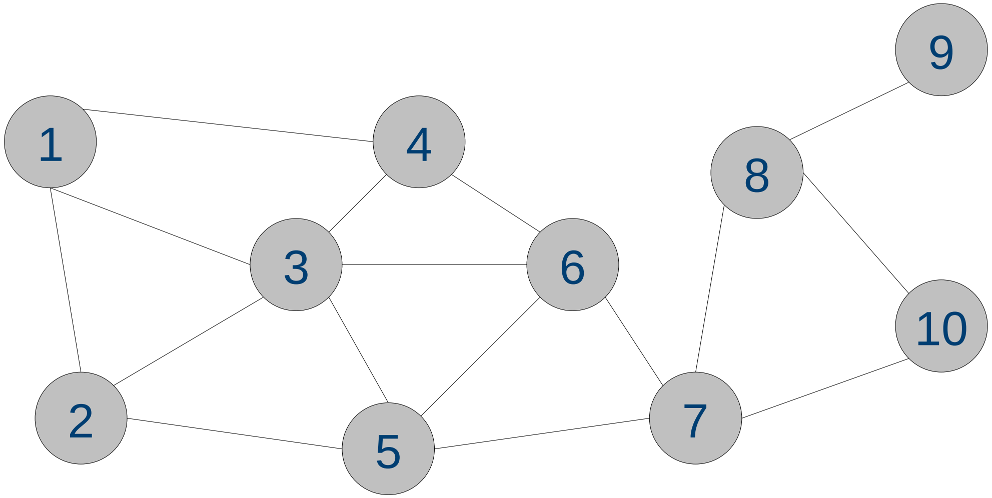
# Shortest Paths

- Social networks shown to be “small world”
- Short paths should exist, even for large graphs
- Short paths can be used for social engineering

# Floyd-Warshall Algorithm

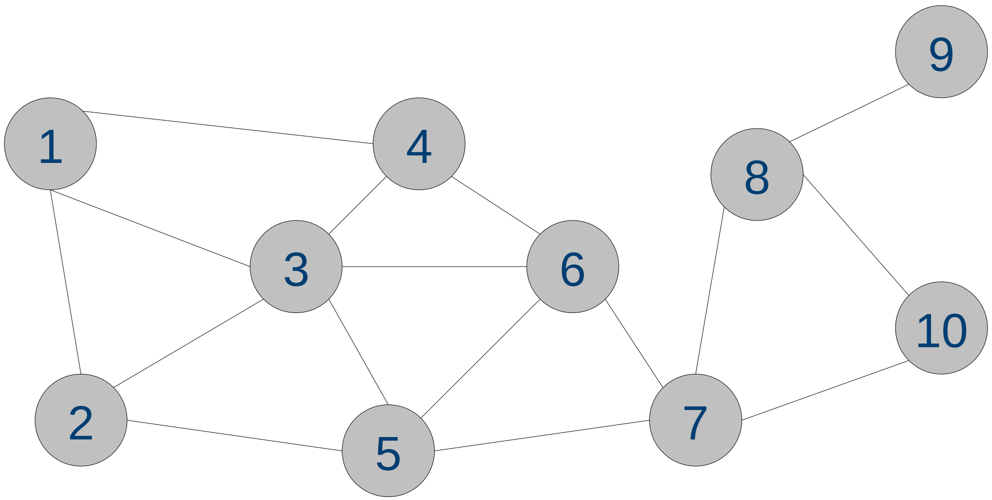
- Finds shortest distance between all pairs of nodes
- Dynamic programming – O($V^3$) over $V^2$ nodes
- Think Dijkstra, but for all vertices

# Floyd-Warshall Algorithm



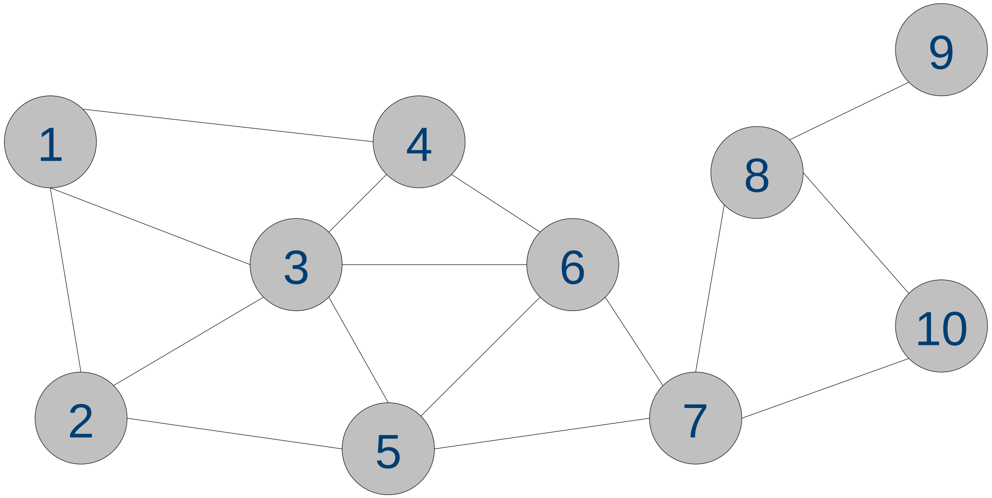

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 1 | 1 | 1 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **2** | 1 | 0 | 1 | ∞ | 1 | ∞ | ∞ | ∞ | ∞ | ∞ |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 | ∞ | ∞ | ∞ | ∞ |
| **4** | 1 | ∞ | 1 | 0 | ∞ | 1 | ∞ | ∞ | ∞ | ∞ |
| **5** | ∞ | 1 | 1 | ∞ | 0 | 1 | 1 | ∞ | ∞ | ∞ |
| **6** | ∞ | ∞ | 1 | 1 | 1 | 0 | 1 | ∞ | ∞ | ∞ |
| **7** | ∞ | ∞ | ∞ | ∞ | 1 | 1 | 0 | 1 | ∞ | 1 |
| **8** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 1 | 0 | 1 | 1 |
| **9** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 1 | 0 | ∞ |
| **10** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 1 | 1 | ∞ | 0 |

# Floyd-Warshall Algorithm



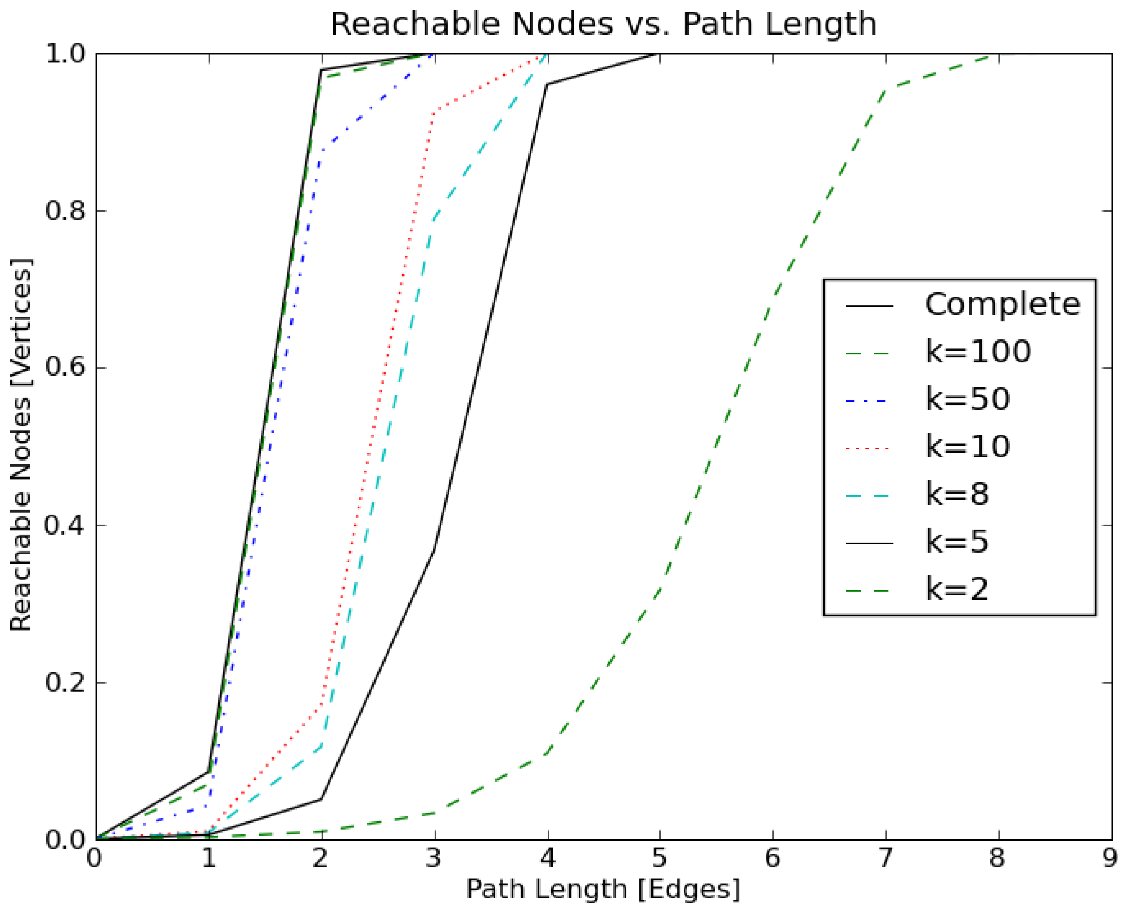

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 1 | 1 | 1 | 2 | 2 | ∞ | ∞ | ∞ | ∞ |
| **2** | 1 | 0 | 1 | 2 | 1 | 2 | 2 | ∞ | ∞ | ∞ |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 | 2 | ∞ | ∞ | ∞ |
| **4** | 1 | 2 | 1 | 0 | 2 | 1 | 2 | ∞ | ∞ | ∞ |
| **5** | 2 | 1 | 1 | 2 | 0 | 1 | 1 | 2 | ∞ | 2 |
| **6** | 2 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | ∞ | 2 |
| **7** | ∞ | 2 | 2 | 2 | 1 | 1 | 0 | 1 | 2 | 1 |
| **8** | ∞ | ∞ | ∞ | ∞ | 2 | 2 | 1 | 0 | 1 | 1 |
| **9** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 2 | 1 | 0 | 2 |
| **10** | ∞ | ∞ | ∞ | ∞ | 2 | 2 | 1 | 1 | 2 | 0 |

# Floyd-Warshall Algorithm




| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 4 |
| **2** | 1 | 0 | 1 | 2 | 1 | 2 | 2 | 3 | 4 | 3 |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 | 2 | 3 | 4 | 3 |
| **4** | 1 | 2 | 1 | 0 | 2 | 1 | 2 | 3 | 4 | 3 |
| **5** | 2 | 1 | 1 | 2 | 0 | 1 | 1 | 2 | 3 | 2 |
| **6** | 2 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | 3 | 2 |
| **7** | 3 | 2 | 2 | 2 | 1 | 1 | 0 | 1 | 2 | 1 |
| **8** | 4 | 3 | 3 | 3 | 2 | 2 | 1 | 0 | 1 | 1 |
| **9** | 5 | 4 | 4 | 4 | 3 | 3 | 2 | 1 | 0 | 2 |
| **10** | 4 | 3 | 3 | 3 | 2 | 2 | 1 | 1 | 2 | 0 |

# Short Paths Still Exist in Sampled Graph



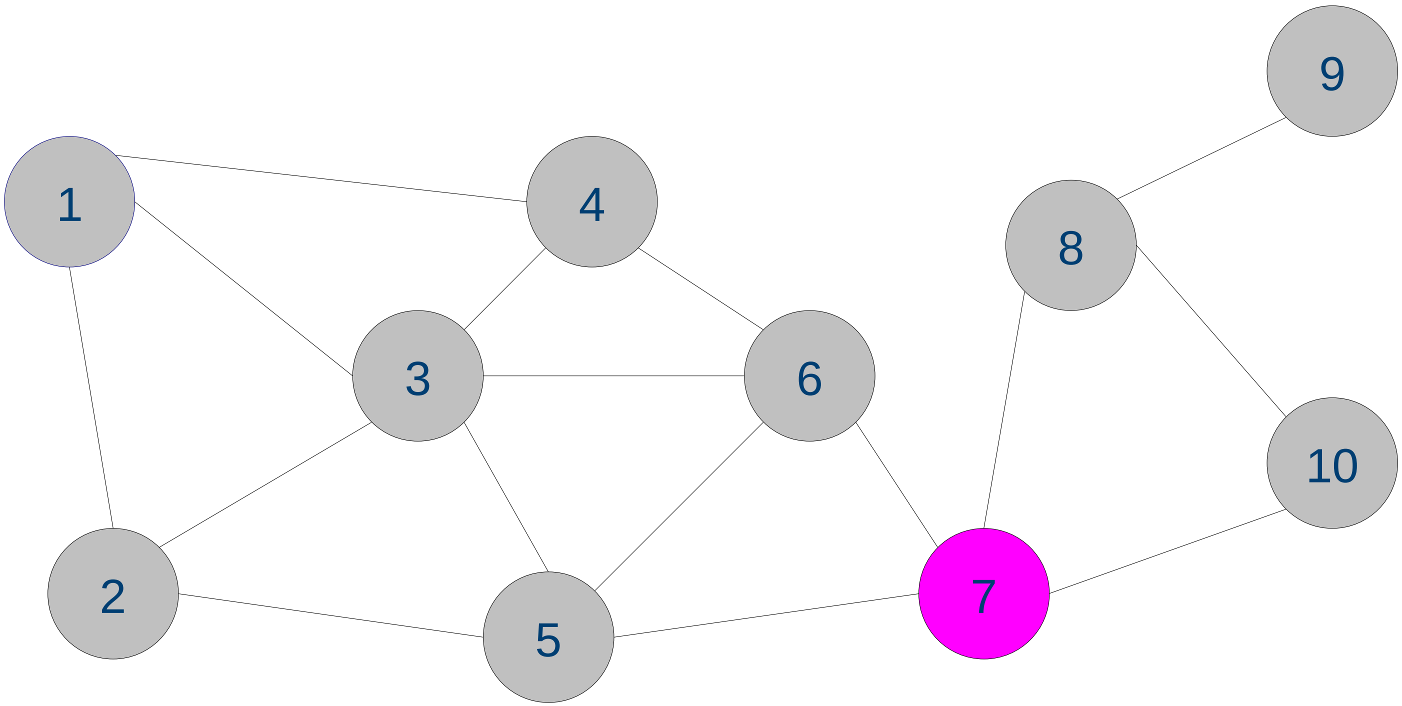
# Centrality

- A measure of a node's importance
- *Betweenness centrality*:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

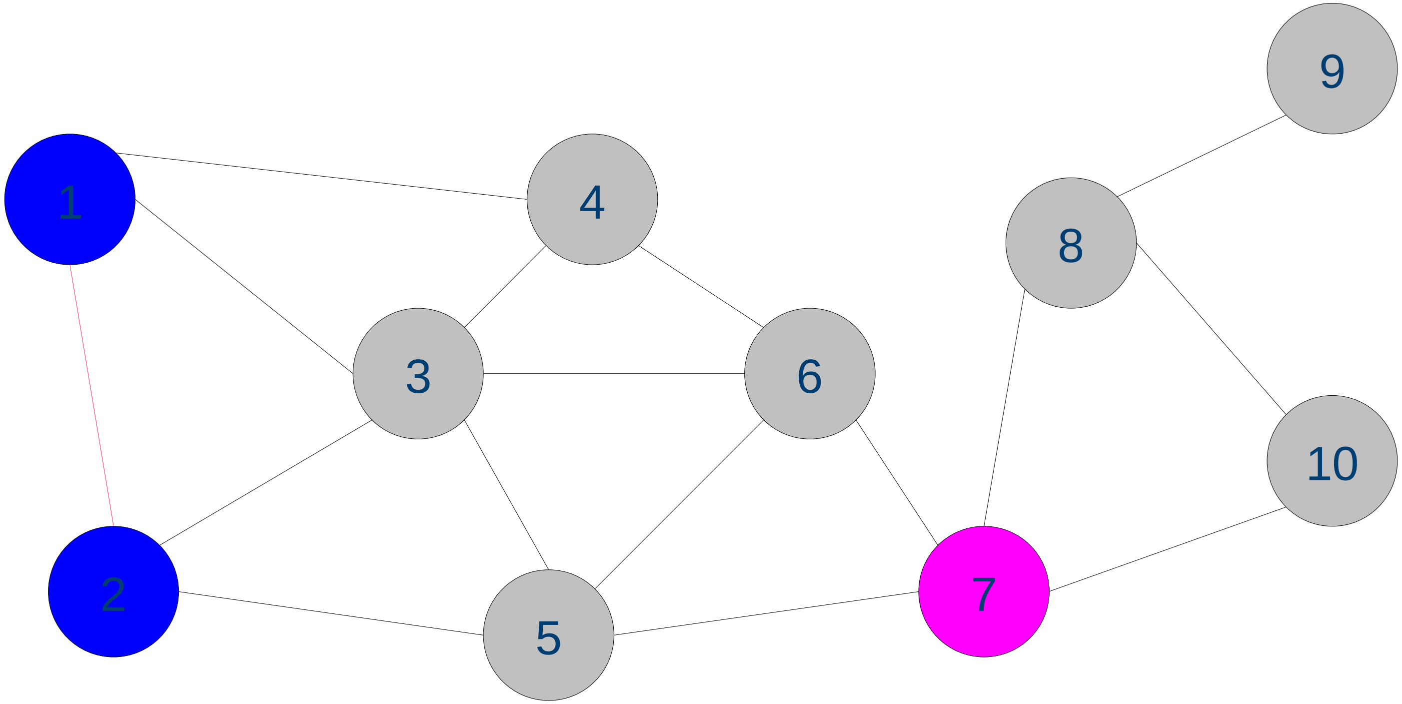
- Measures the shortest paths in the graph that a particular vertex is part of

# Centrality



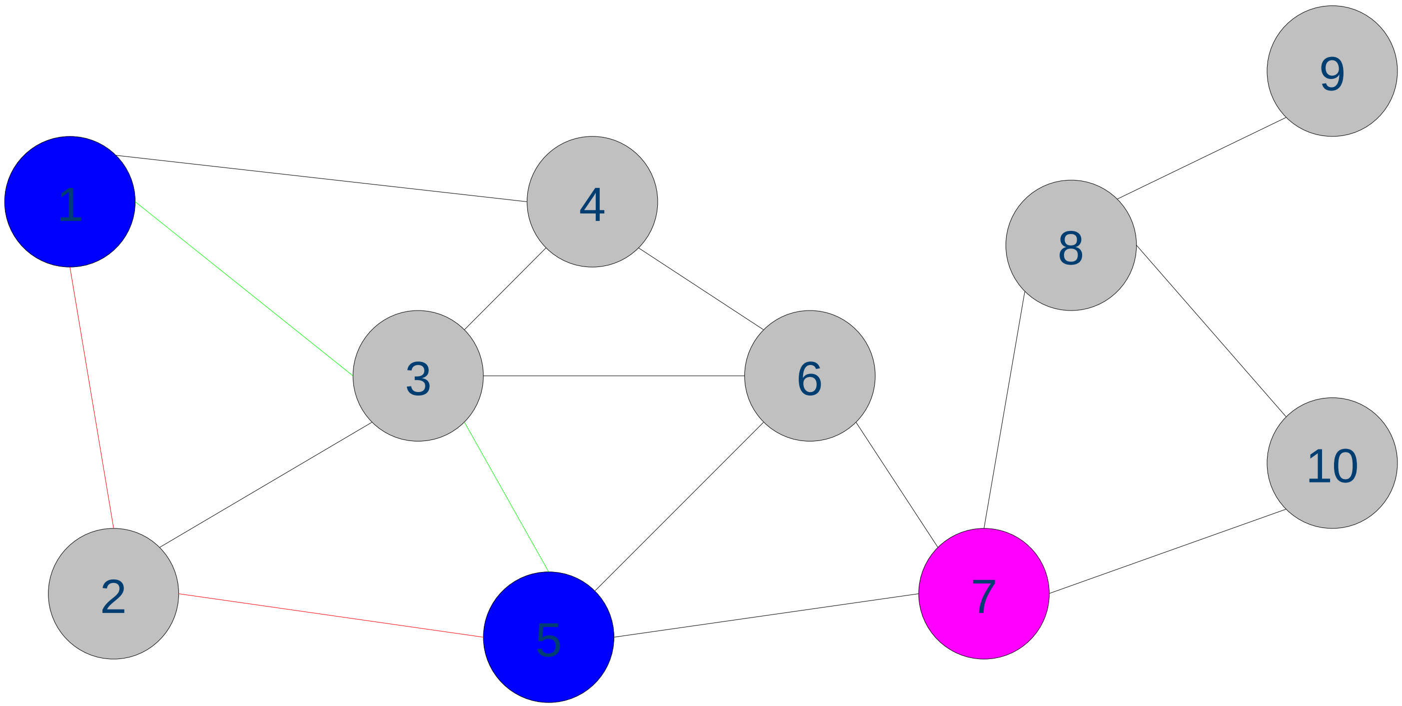

$C_B(v_7) = ?$

# Centrality



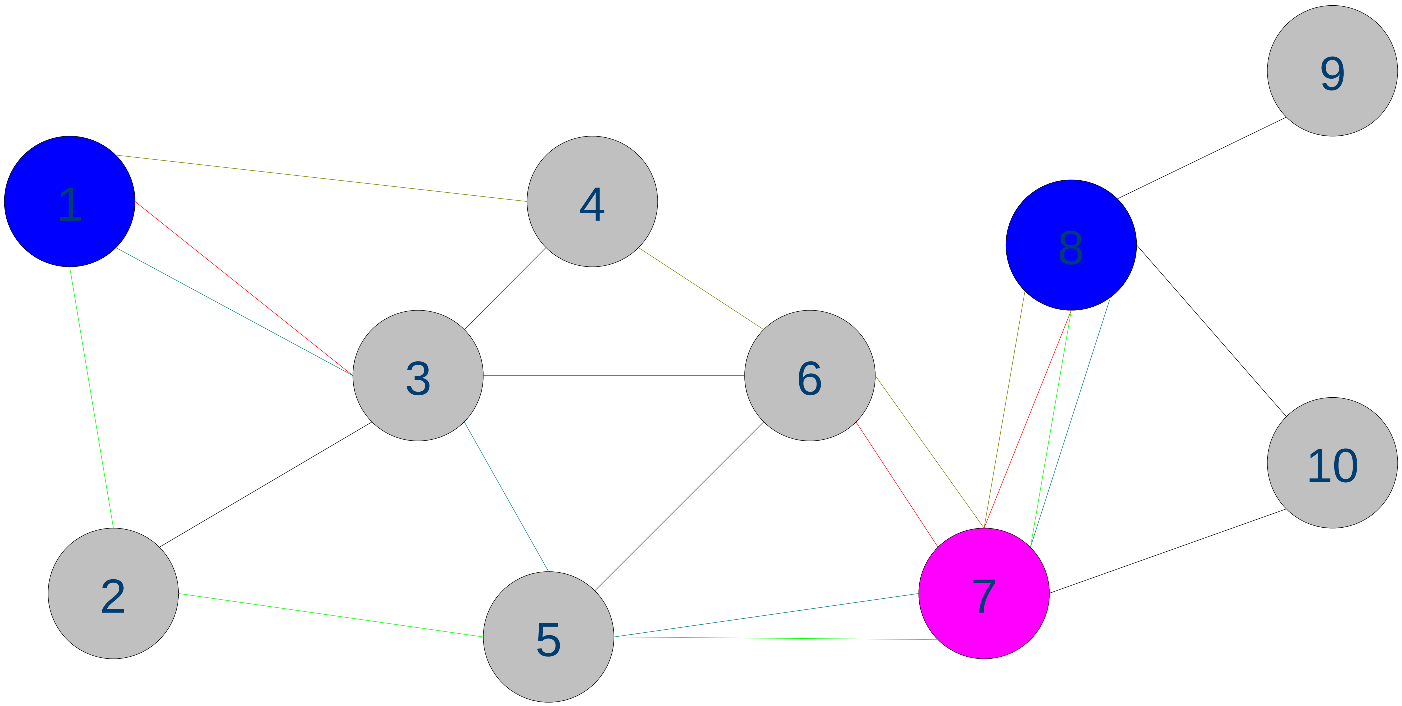

$$C_B(v_7) = \frac{0}{1} +$$

# Centrality




$$C_B(v_7) = \frac{0}{1} + \frac{0}{2} +$$

# Centrality



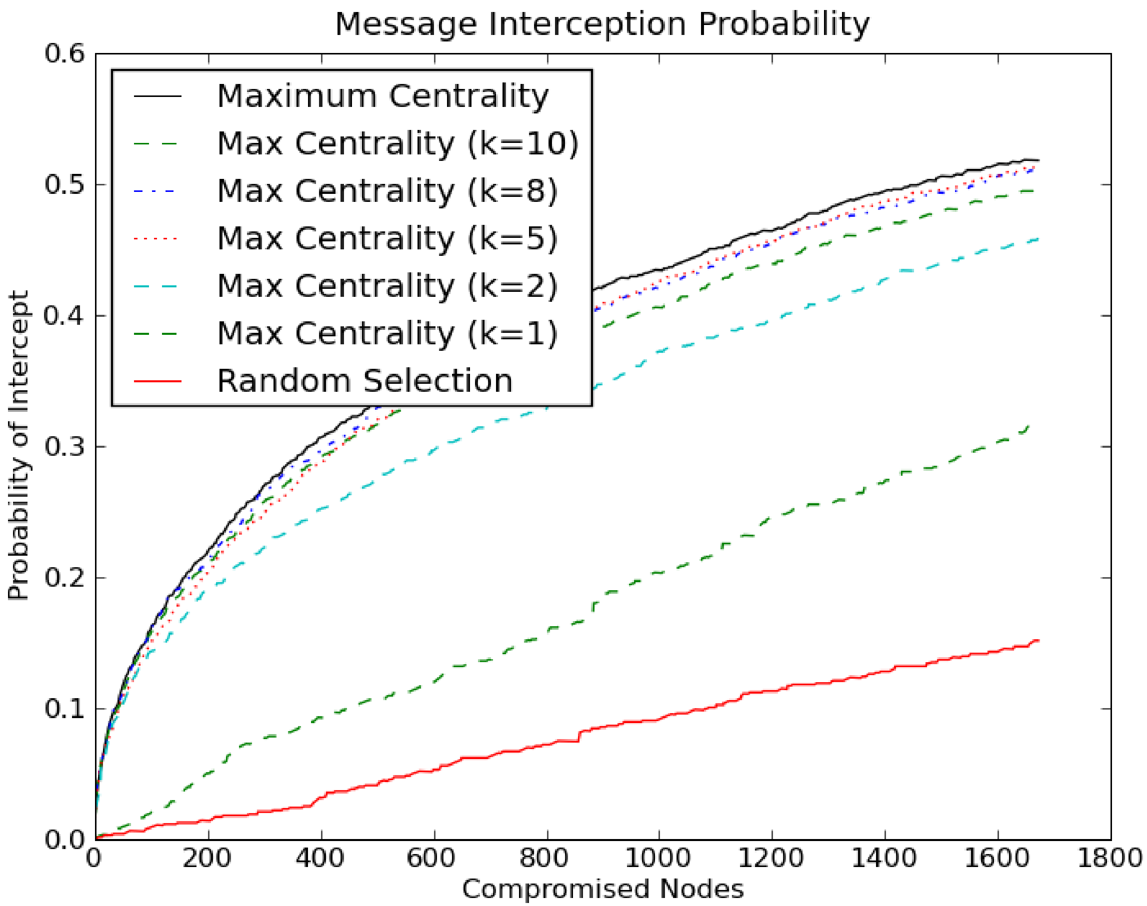

$$C_B(v_7) = \frac{0}{1} + \frac{0}{2} + \frac{4}{4} +$$

# Message Interception Scenario

- Messages sent via shortest (least-cost) paths
- Adversary can compromise *x* nodes
- How much traffic can s/he intercept?

$$p_{intercept}(v_s, v_d) = \frac{C_B(v)}{|V|^2}$$

# Message Interception



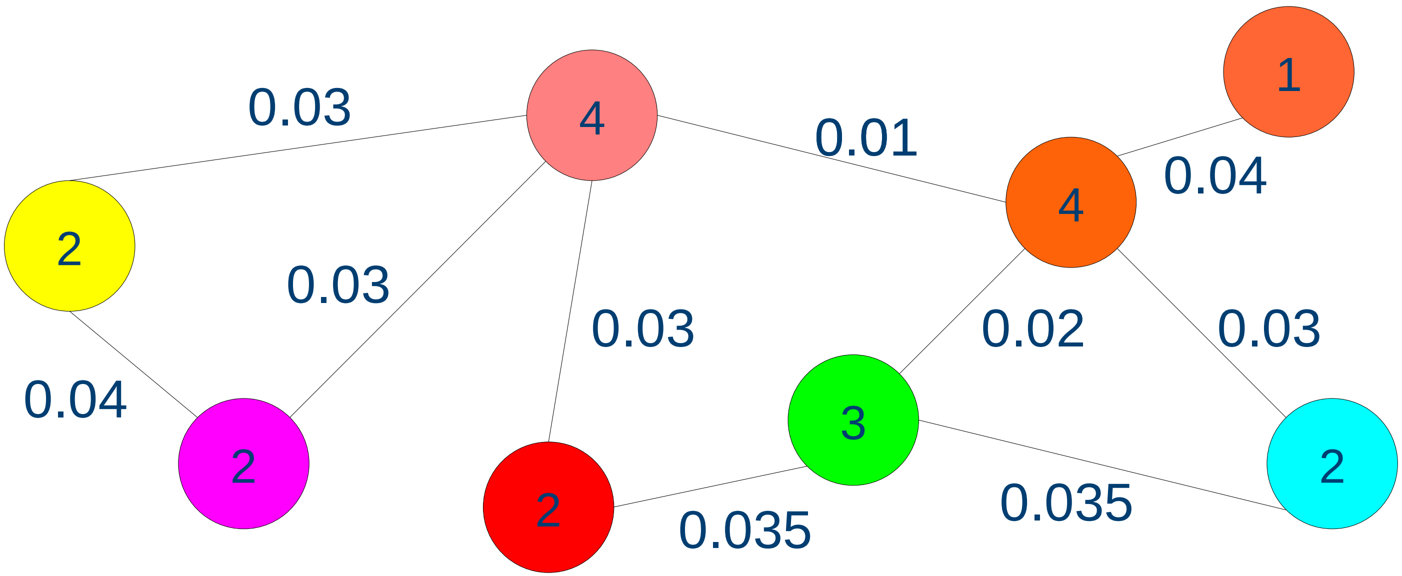
# Community Detection

- Goal: Find highly-connected sub-groups
- Measure success by high *modularity*:

$$Q = \frac{1}{2m} \sum_{v,w} \left[ A_{vw} - \frac{d(v)d(w)}{2m} \right]$$

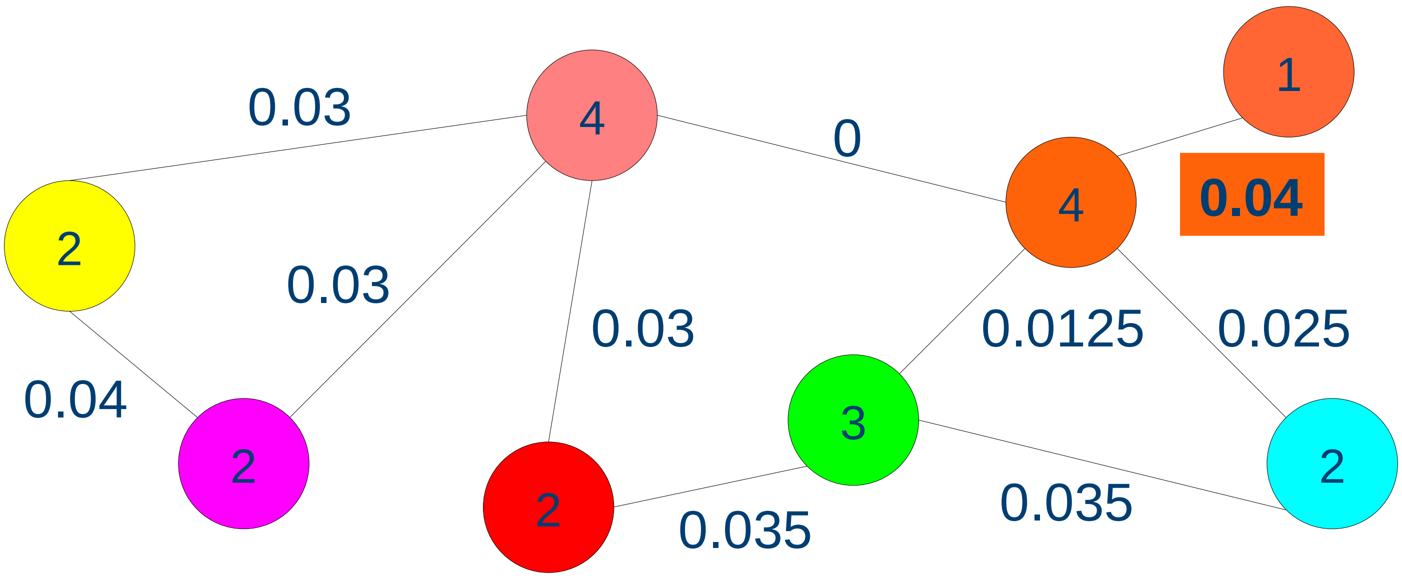
- Ratio of intra-community edges to random
- Normalised to be between -1 and 1

# Community Detection



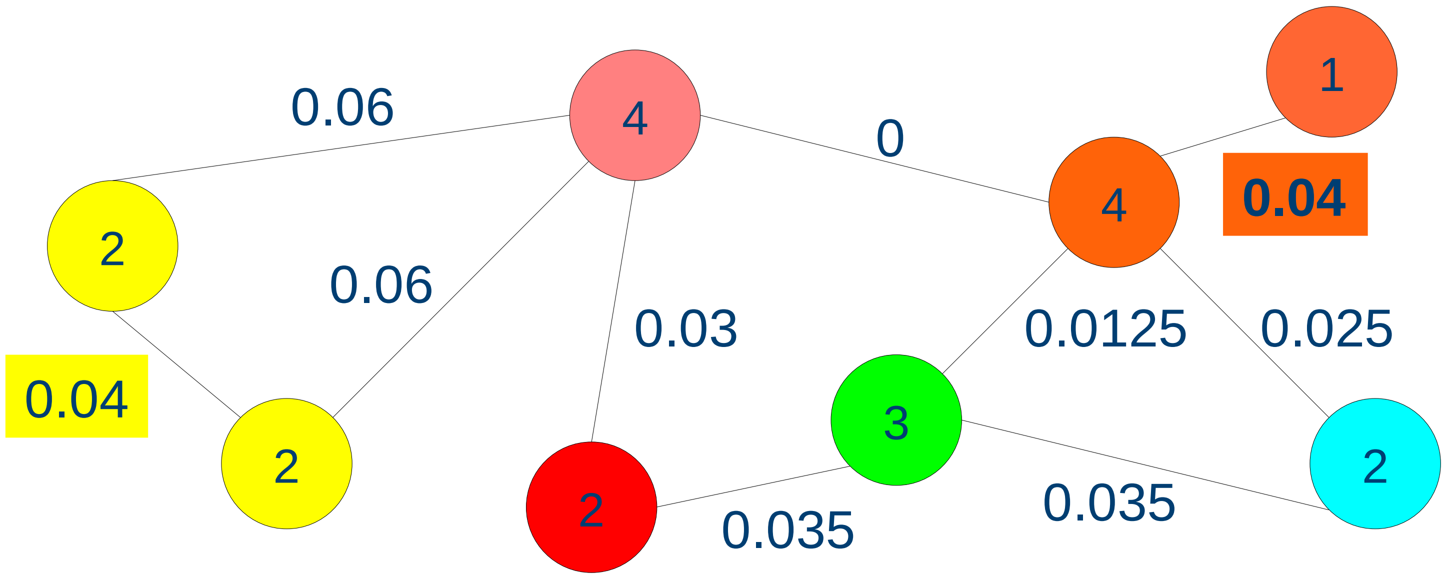

- Clausen et. al 2004 – find maximal modularity in O($n\lg^2 n$)
- Track marginal modularity, update neighbours on each merge

# Community Detection



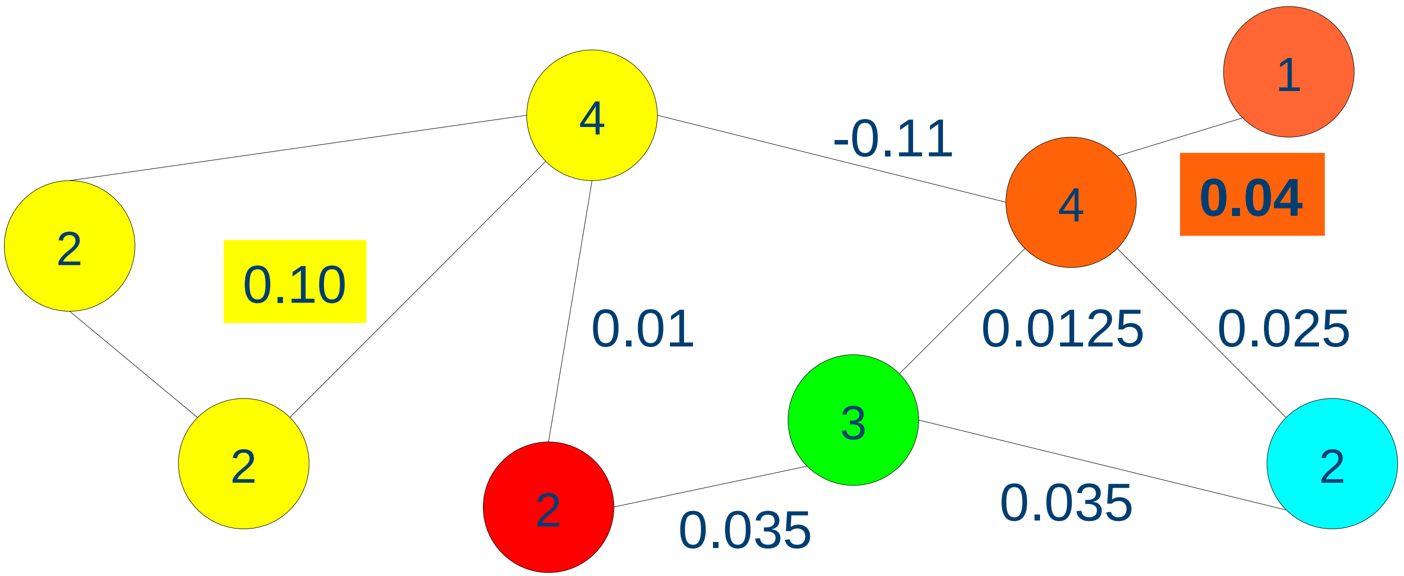

Q=0.04

# Community Detection



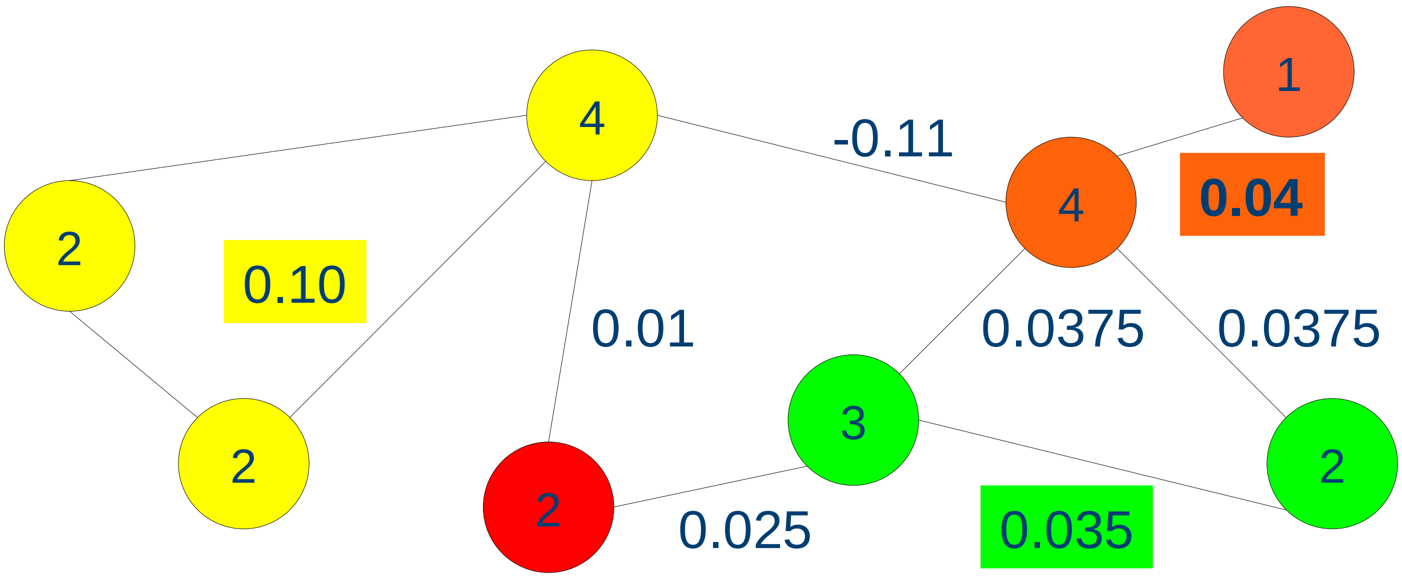

Q=0.08

# Community Detection



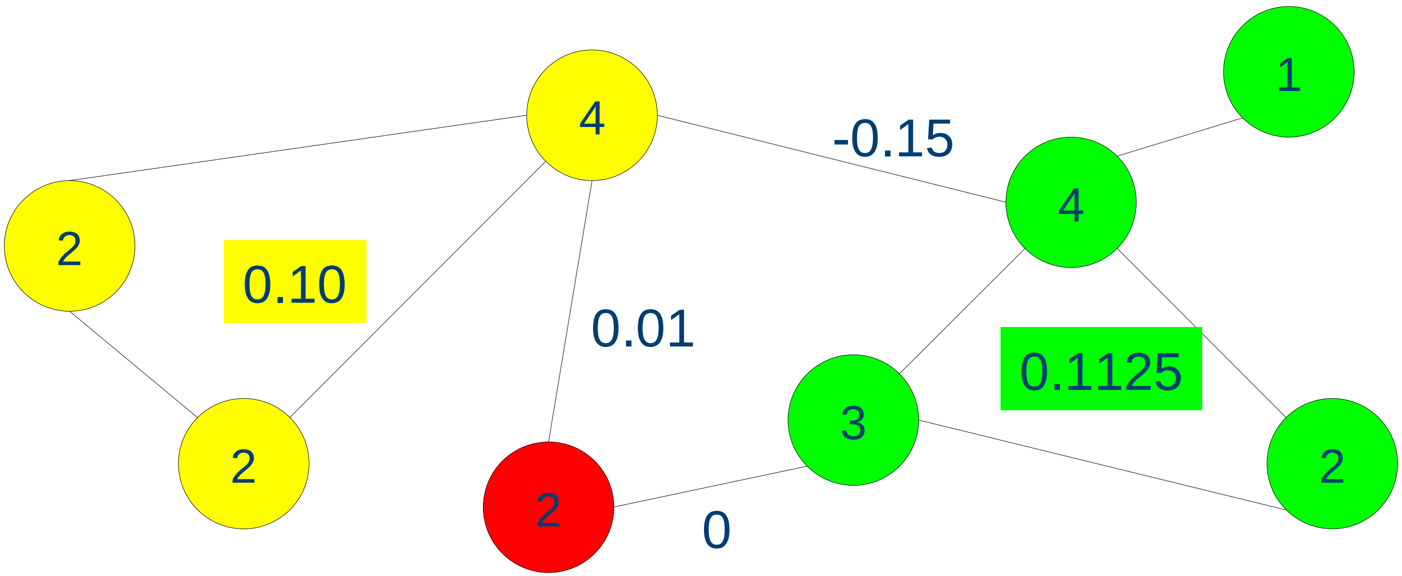

Q=0.14

# Community Detection



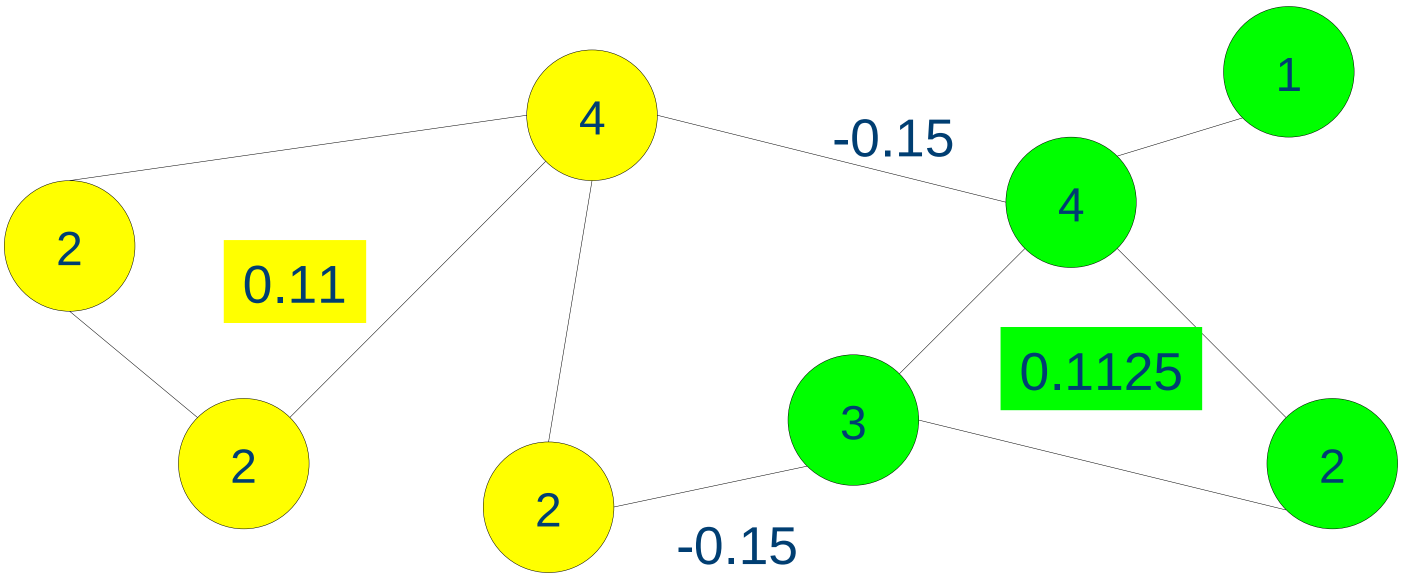

Q=0.175

# Community Detection



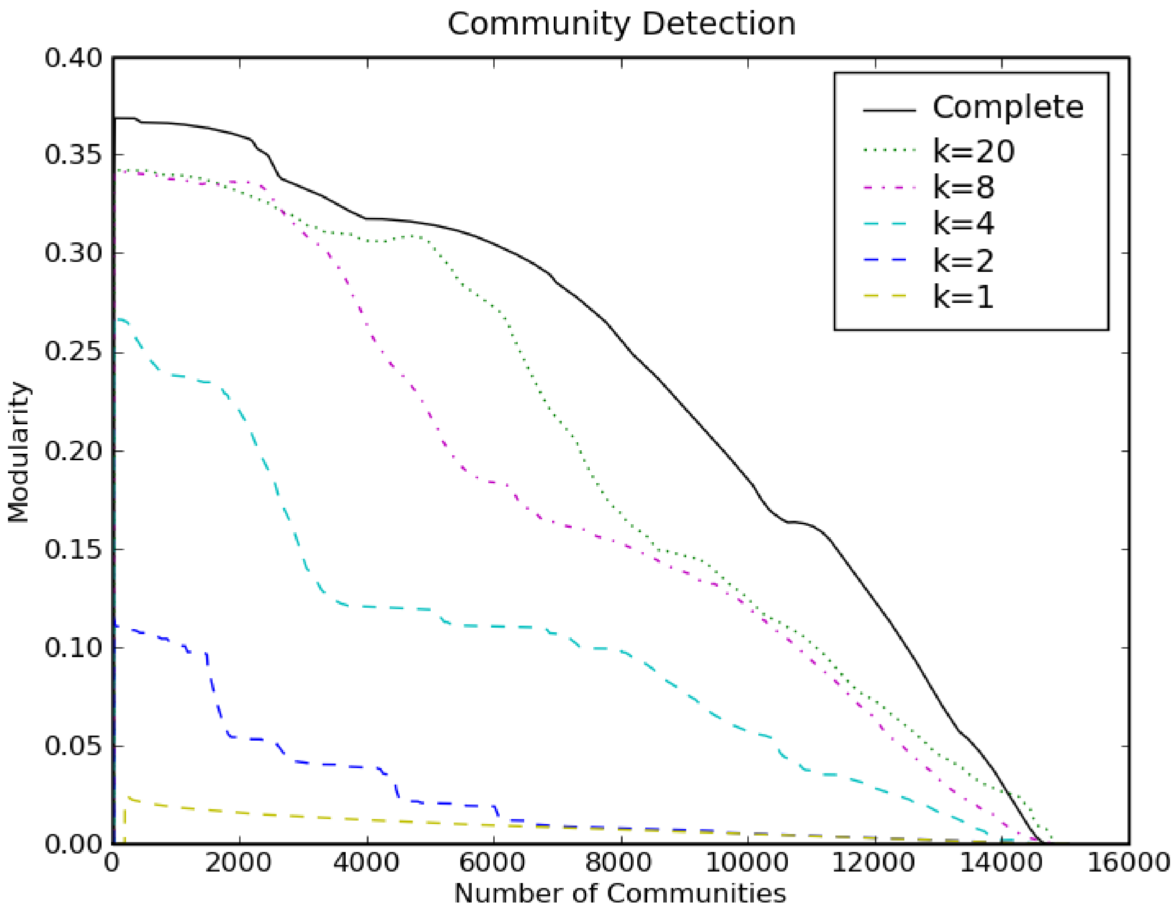

Q=0.2125

# Community Detection




Q=0.2225

# Community Detection



# Conclusions

- Social graph is fragile to partial disclosure
  - Consistent with Danezis/Wittneben, Nagaraja results
- Public Listings Leak Too Much
  - Dominating sets, centrality, communities in particular
- SNS operators need a dedicated privacy review team
  - Comparable to security audit & penetration testing

# Questions?

jcb82@cl.cam.ac.uk

jra40@cl.cam.ac.uk