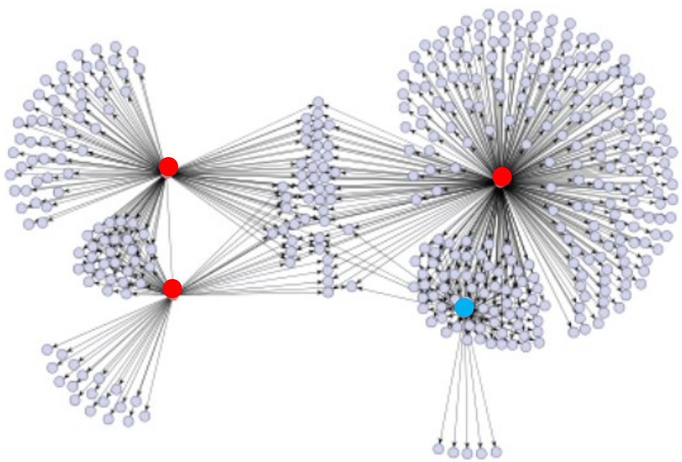
LECTURE 10:INFLUENCE MAXIMIZATION IN NETWORKS

COMP4641: Social Information Network Analysis and Engineering

Friday March 20nd 2015

How to Create Big Cascades?

- **Blogs – Information epidemics:**
  - Which are the influential blogs?
  - Which blogs create big cascades?
  - Where should we advertise?

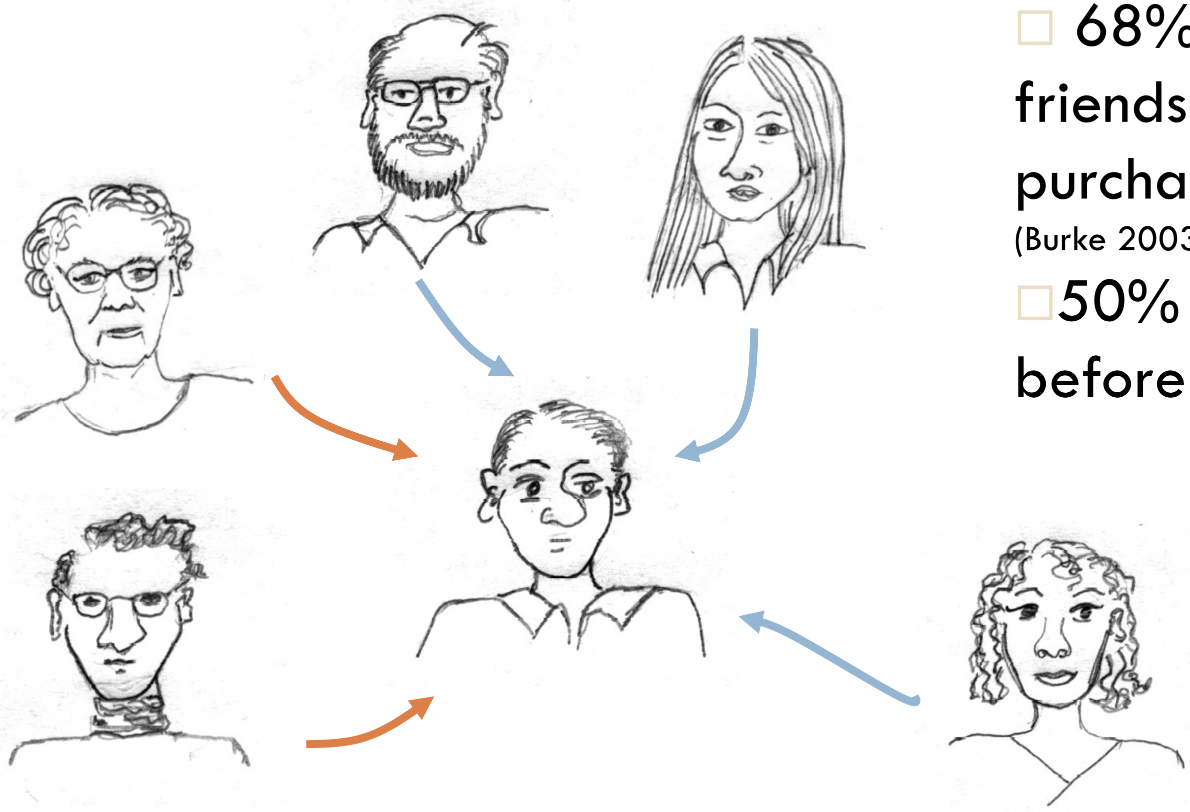


Which node shall we target?

● vs. ●

Viral Marketing?

- **We are more influenced by our friends than strangers**



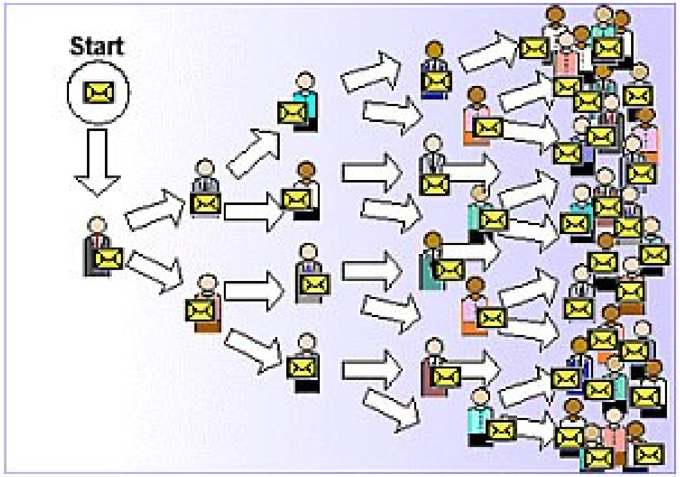
- 68% of consumers consult friends and family before purchasing home electronics (Burke 2003)
- 50% do research online before purchasing electronics

Viral Marketing

Identify influential customers

Convince them to adopt the product – Offer discount/free samples

These customers endorse the product among their friends

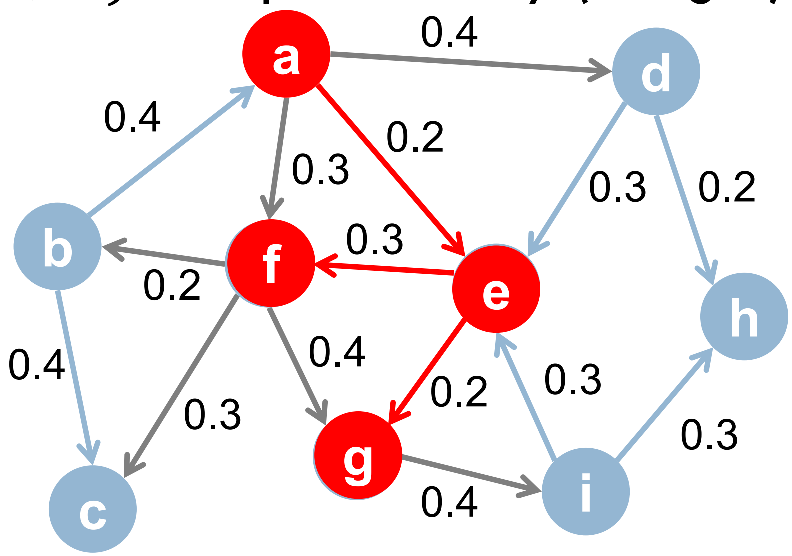


Probabilistic Contagion

- **Independent Cascade Model**
  - Directed finite $G = (V, E)$
  - Set S starts out with new behavior
    - Say nodes with this behavior are “active”
  - Each edge (v, w) has a probability p_{vw}
  - If node v is active, it gets <u>one</u> chance to make w active, with probability p_{vw}
    - Each edge fires at most once
- **Does scheduling matter? No**
    - u, v both active, doesn’t matter which fires first
  - **But the time moves in discrete steps**

Independent Cascade Model

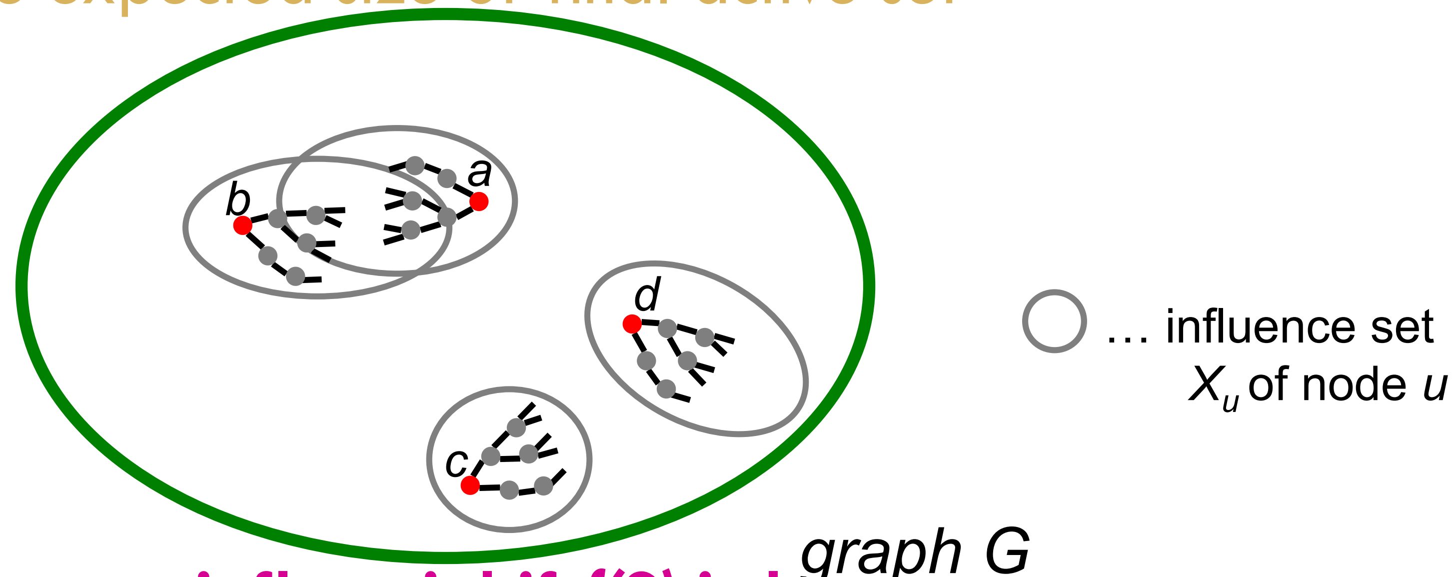
- **Initially some nodes S are active**
- Each edge (v, w) has probability (weight) p_{vw}



- **When node v becomes active:**
  - It activates each out-neighbor w with prob. p_{vw}
- **Activations spread through the network**

Most Influential Set of Nodes

- ***S***: is initial active set
- ***f(S)***: The expected size of final active set



- **Set S is more influential if *f(S)* is larger**

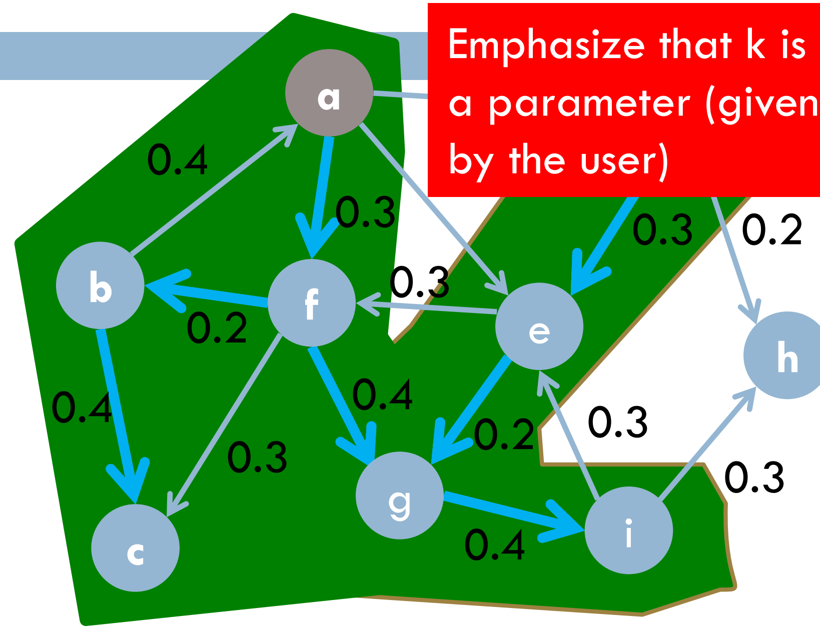
$$f(\{a,b\}) < f(\{a,c\}) < f(\{a,d\})$$

Most Influential Set

Emphasize that k is a parameter (given by the user)

Problem:

- **Most influential set of size *k***: set S of k nodes producing largest **expected cascade size *f(S)*** if activated [Domingos-Richardson '01]



Influence set X_a of a

Influence set X_d of d

- **Optimization problem:**

$$\max_{\text{S of size k}} f(S)$$

Why "expected cascade size"? X_a is a result of a random process. So in practice we would want to compute many realizations of X_a and then maximize the avg. $f(S)$

$$f(S) = \sum_{\substack{\text{Random} \\ \text{realizations } i}} f_i(S)$$

HOW HARD IS INFLUENCE MAXIMIZATION?

Most Influential Subset of Nodes

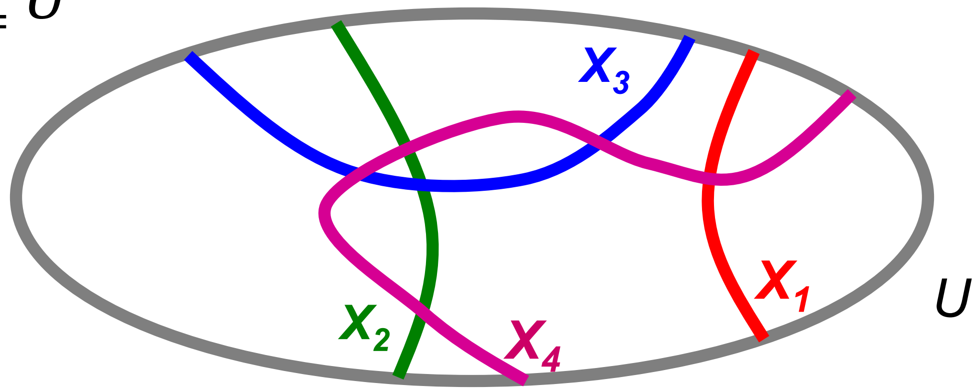
- **Most influential set of *k* nodes:** set S on k nodes producing largest expected cascade size $f(S)$ if activated
- **The optimization problem:**

$$\max_{S \text{ of size } k} f(S)$$

- **How hard is this problem?**
  - **NP-COMPLETE!**
    - Show that finding most influential set is at least as hard as a vertex cover

Background: Vertex Cover

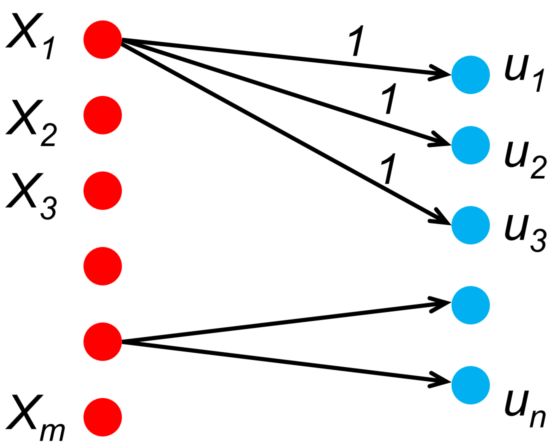
- **Vertex cover problem**
  (a known NP-complete problem):
  - Given universe of elements $U = \{u_1, \ldots, un\}$ and sets $X_1, \ldots, X_m \subseteq U$



  - **Are there *k* sets among *$X_1, \ldots, X_m$* such that their union is *U*?**
- **Goal:**
  Encode vertex cover as an instance of $\max_{S \text{ of size } k} f(S)$

Influence Maximization is NP-hard

- **Given a vertex cover instance with sets $X_1,\ldots, X_m$**
- **Build a bipartite "X-to-U" graph:**



e.g.:
$X_1 = \{u_1, u_2, u_3\}$

Construction:
- Create edge (X_i,u) $\forall X_i$ $\forall u \in X_i$ -- directed edge from sets to their elements
- Put weight 1 on each edge (*e.i.*, activation is deterministic)

- **Vertex cover as Influence Maximization in X-to-U graph: There exists a set *S* of size *k* with *f(S)=k+n* iff there exists a size *k* set cover**

Note: Optimal solution is always a set of sets X_i.
This problem is hard in general, could be special cases that are easier.

Summary so Far

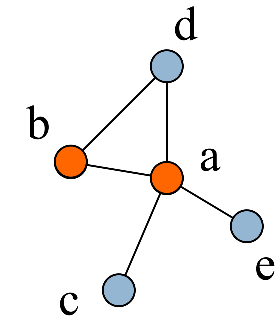
- Bad news:
  - Influence maximization is NP-complete
- Next, good news:
  - There exists an approximation algorithm!
- Consider the Hill Climbing algorithm to find S:
  - **Input:**
    Influence set of each node u: $X_u = \{v_1, v_2, \dots\}$
    - If we activate u, nodes $\{v_1, v_2, \dots\}$ will eventually get active
  - **Algorithm:** At each iteration i take the node u that gives best marginal gain: $\max_u f(S_{i-1} \cup \{u\})$

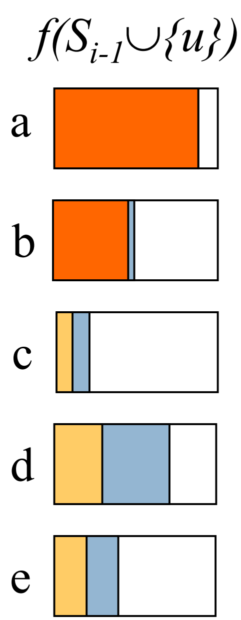
S_i … Initially active set
$f(S_i)$ … Size of the union of X_u, $u \in S_i$

(Greedy) Hill Climbing

Algorithm:

- Start with $S_0 = \{\,\}$
- For $i = 1 \ldots k$
  - Take node u that $\max f(S_{i-1} \cup \{u\})$
  - Let $S_i = S_{i-1} \cup \{u\}$





- **Example:**
  - Eval. $f(\{a\}), \ldots, f(\{e\})$, pick max of them
  - Eval. $f(\{a,b\}), \ldots, f(\{a,e\})$, pick max
  - Eval. $f(a,b,c\}), \ldots, f(\{a,b,e\})$, pick max

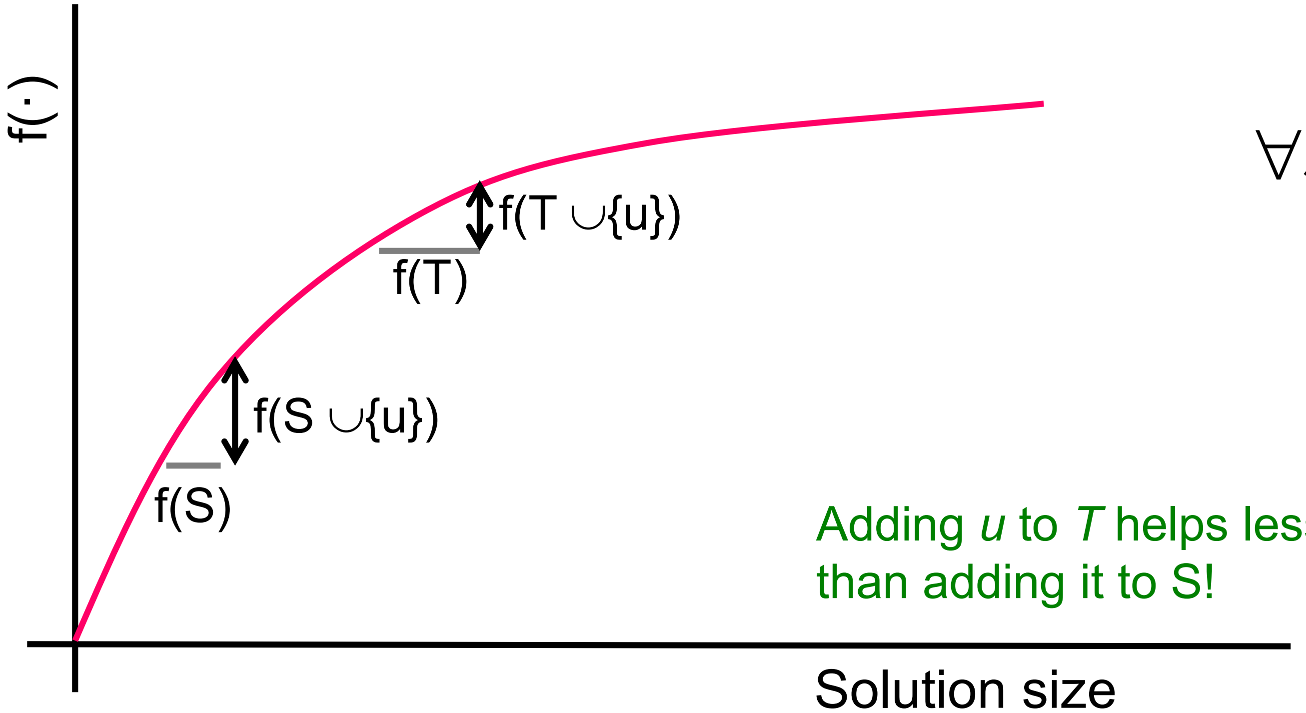
Approximation Guarantee

- **Hill climbing produces a solution S where: f(S) ≥(1-1/e)*OPT** (f(S)>0.63*OPT)
  [Nemhauser, Fisher, Wolsey '78, Kempe, Kleinberg, Tardos '03]
- **Claim holds for functions *f(·)* with 2 properties:**
  - ***f* is monotone:** (activating more nodes doesn't hurt)
    if $S \subseteq T$ then $f(S) \leq f(T)$ and $f(\{\})=0$
  - ***f* is submodular:** (activating each additional node helps less)
    adding an element to a set gives less improvement than adding it to one of its subsets: $\forall \boldsymbol{S} \subseteq \boldsymbol{T}$

$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

Submodularity– Diminishing returns

- **Diminishing returns:**



$$\forall S \subseteq T$$

$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

Solution Quality

We just proved:

- Hill climbing finds solution *S* which
  f(S) ≥ (1-1/e)*OPT i.e., f(S) ≥ 0.63*OPT

- This is a **data independent bound**
  - This is a worst case bound
  - No matter what is the input data (influence sets), we know that the Hill-Climbing won't never do worse than 0.63*OPT

Evaluating $f(S)$?

- **How to evaluate $f(S)$?**
  - Still an open question of how to compute efficiently
- **But:** Very good estimates by simulation
  - Repeating the diffusion process often enough (polynomial in n; $1/\varepsilon$)
  - Achieve $(1\pm\varepsilon)$-approximation to $f(S)$
  - Generalization of Nemhauser-Wolsey proof: Greedy algorithm is now a $(1-1/e-\varepsilon')$-approximation

SIMULATION EXPERIMENTS

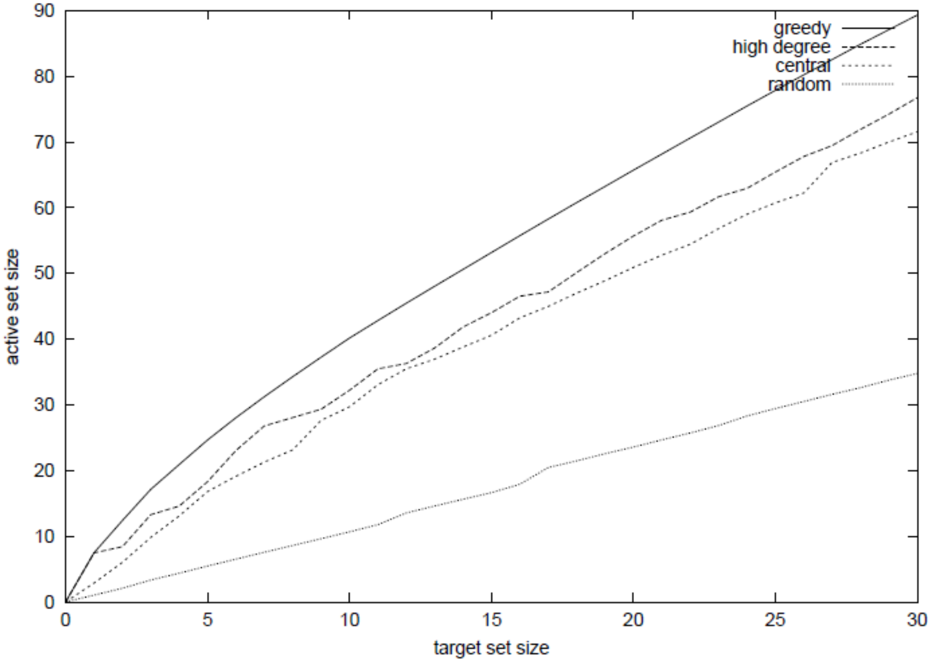
Experiment Data

- **A collaboration network:** co-authorships in papers of the arXiv high-energy physics theory:
  - 10,748 nodes
  - 53,000 edges
- **Independent Cascade Model:**
  - **Case 1:** Uniform probabilities p on each edge
  - **Case 2:** Edge from v to ω has probability 1/ deg(ω) of activating ω.

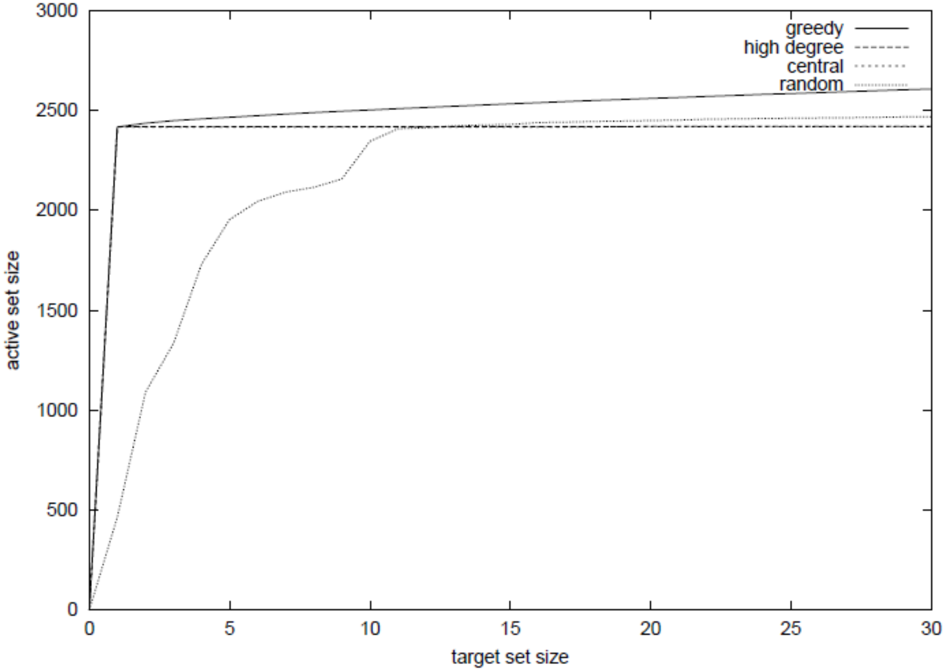
Experiment Settings

- **Simulate the process 10,000 times for each targeted set**
  - Every time re-choosing edge outcomes randomly
- **Compare with other 3 common heuristics**
  - Degree centrality,
  - Distance centrality
  - Random nodes

Independent Cascade Model

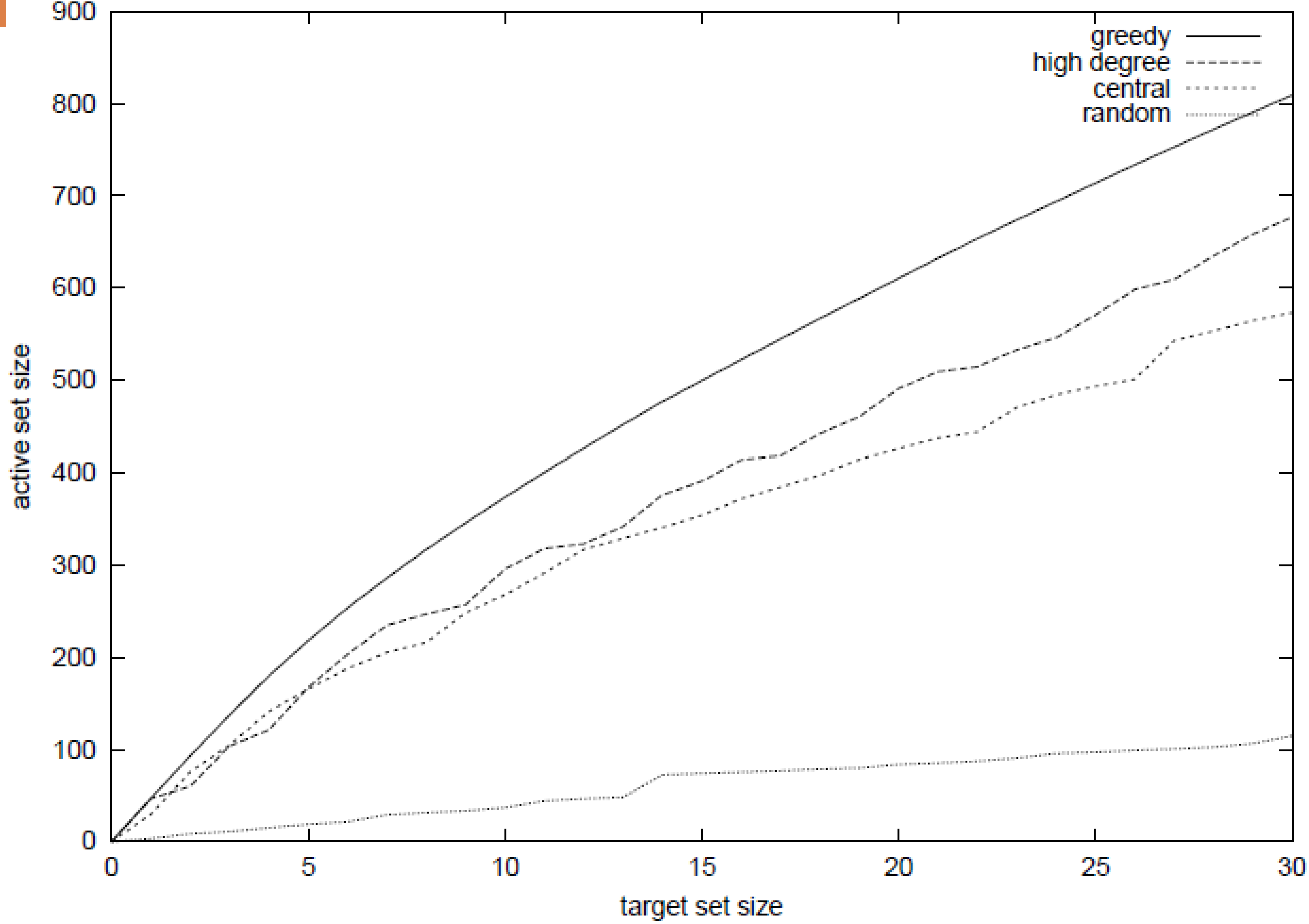


$p_{uv} = 1\%$



$p_{uv} = 10\%$

Independent Cascade Model



$p_{uv}=1/deg(v)$