Maximizing the Spread of Influence through a Social Network

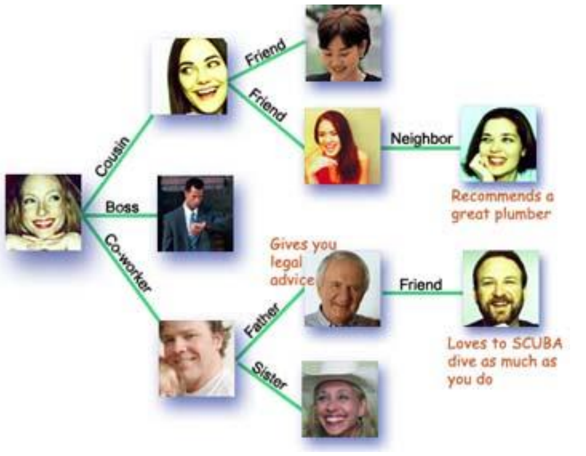
Authors: David Kempe, Jon Kleinberg, Éva Tardos

KDD 2003

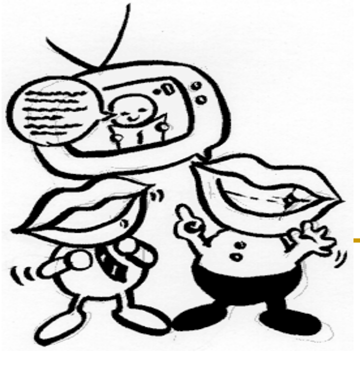
Adapted from author's slide at:
http://www.cs.washington.edu/affiliates/meetings/talks04/kempe.pdf

Social Network and Spread of Influence

- Social network plays a fundamental role as a medium for the spread of INFLUENCE among its members
  - Opinions, ideas, information, innovation…



- Direct Marketing takes the "word-of-mouth" effects to significantly increase profits (Gmail, Tupperware popularization, Microsoft Origami …)



Problem Setting

- Given
  - a limited budget B for initial advertising (e.g. give away free samples of product)
  - estimates for influence between individuals
- Goal
  - trigger a large cascade of influence (e.g. further adoptions of a product)
- Question
  - Which set of individuals should B target at?
- Application besides product marketing
  - spread an innovation
  - detect stories in blogs

What we need

- Form models of influence in social networks.
- Obtain data about particular network (to estimate inter-personal influence).
- Devise algorithm to maximize spread of influence.

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Models of Influence

- First mathematical models
  - [Schelling '70/'78, Granovetter '78]
- Large body of subsequent work:
  - [Rogers '95, Valente '95, Wasserman/Faust '94]
- Two basic classes of diffusion models: threshold and cascade
- General operational view:
  - A social network is represented as a directed graph, with each person (customer) as a node
  - Nodes start either active or inactive
  - An active node may trigger activation of neighboring nodes
  - Monotonicity assumption: active nodes never deactivate

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Linear Threshold Model

- A node v has random threshold $\theta_v \sim U[0,1]$
- A node v is influenced by each neighbor w according to a *weight* b_{vw} such that

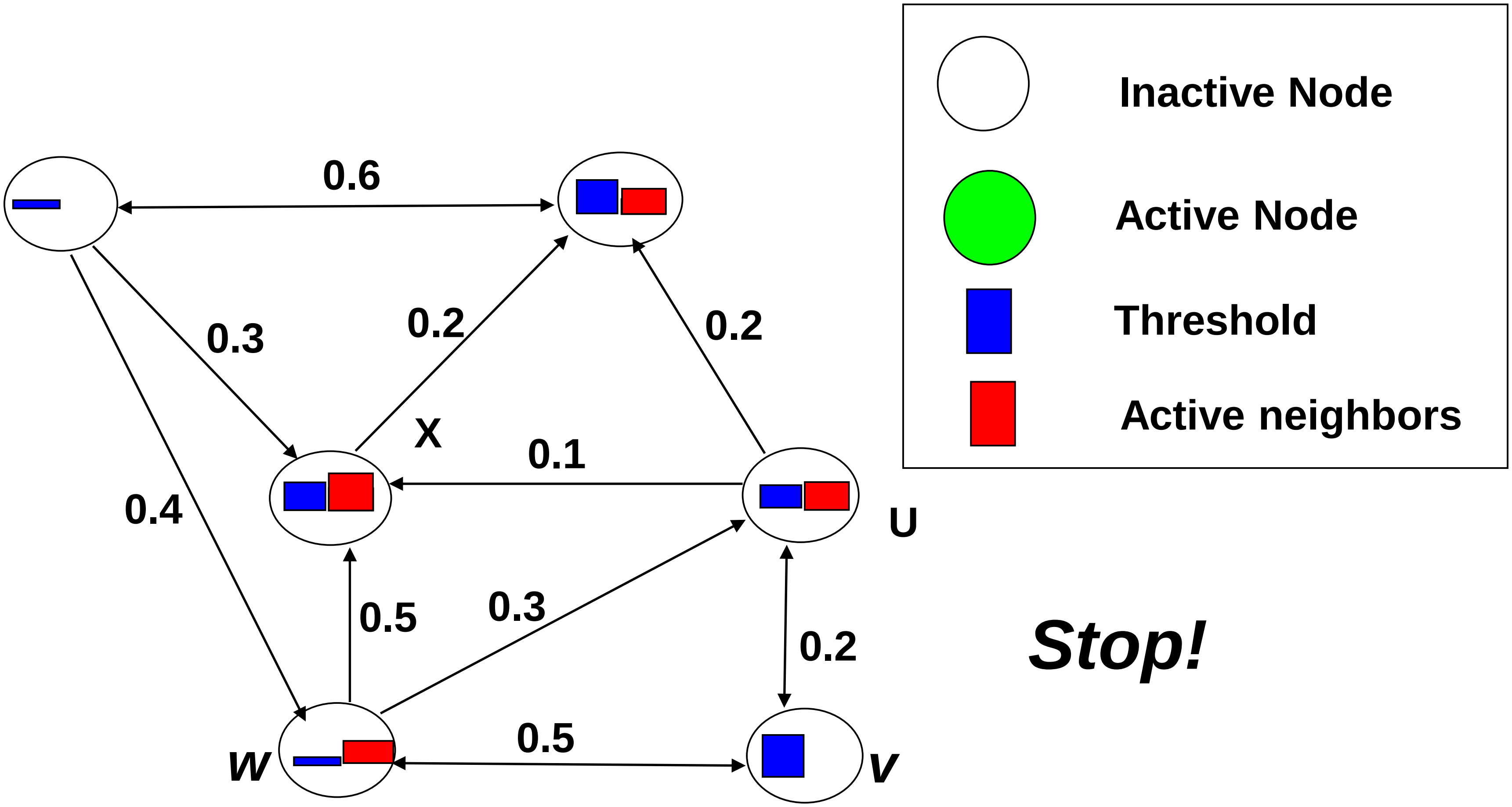
$$\sum_{w \text{ neighbor of } v} b_{v,w} \leq 1$$

- A node v becomes active when at least

(weighted) θ_v fraction of its neighbors are active

$$\sum_{w \text{ active neighbor of } v} b_{v,w} \geq \theta_v$$

Example



Stop!

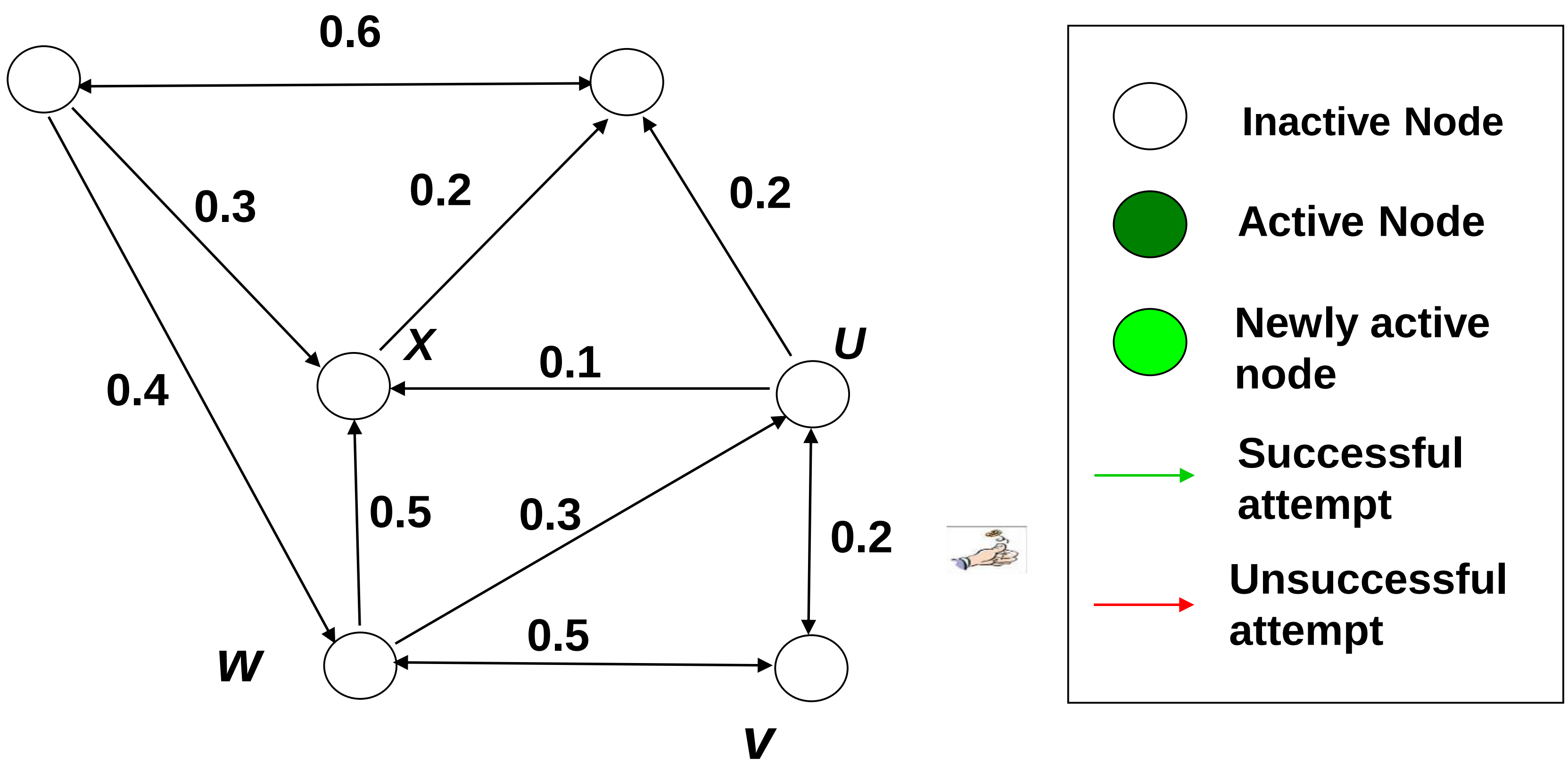
Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Independent Cascade Model

- When node *v* becomes active, it has a single chance of activating each currently inactive neighbor *w*.
- The activation attempt succeeds with probability p_{vw}.

Example



Stop!

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Influence Maximization Problem

- Influence of node set S: f(S)
  - expected number of active nodes at the end, if set S is the initial active set
- Problem:
  - Given a parameter *k* (budget), find a *k*-node set *S* to maximize f(S)
  - Constrained optimization problem with f(S) as the objective function

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

f(S): properties (to be demonstrated)

- Non-negative (obviously)
- Monotone: $f(S+v) \geq f(S)$
- Submodular:
  - Let N be a finite set
  - A set function $f : 2^N \mapsto \Re$ is submodular *iff*

$$\forall S \subset T \subset N, \forall v \in N \setminus T,$$

$$f(S+v) - f(S) \geq f(T+v) - f(T)$$

(diminishing returns)

Bad News

- For a submodular function *f*, if *f* only takes non-negative value, and is monotone, finding a *k*-element set *S* for which *f(S)* is maximized is an NP-hard optimization problem[GFN77, NWF78].
- It is NP-hard to determine the optimum for influence maximization for both independent cascade model and linear threshold model.

Good News

- We can use Greedy Algorithm!
  - Start with an empty set S
  - For k iterations:

    Add node v to S that maximizes $f(S+v) - f(S)$.
- How good (bad) it is?
  - Theorem: The greedy algorithm is a $(1 - 1/e)$ approximation.
  - The resulting set S activates at least $(1- 1/e) > 63\%$ of the number of nodes that any size-k set S could activate.

Outline

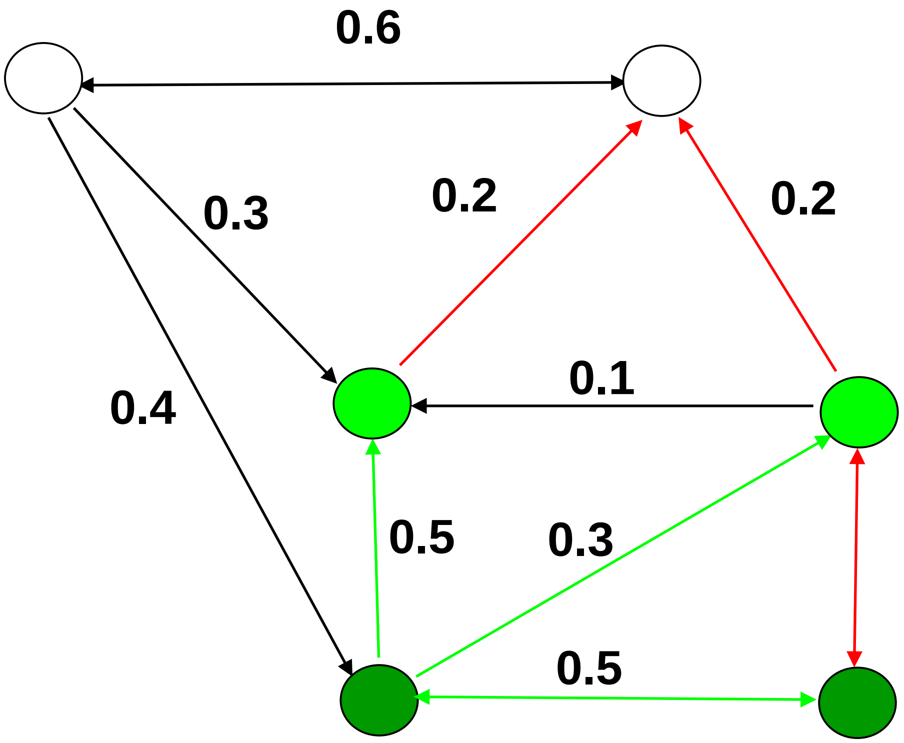
- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Key 1: Prove submodularity

$$\forall S \subset T \subset N, \forall v \in N \setminus T,$$
$$f(S+v) - f(S) \geq f(T+v) - f(T)$$

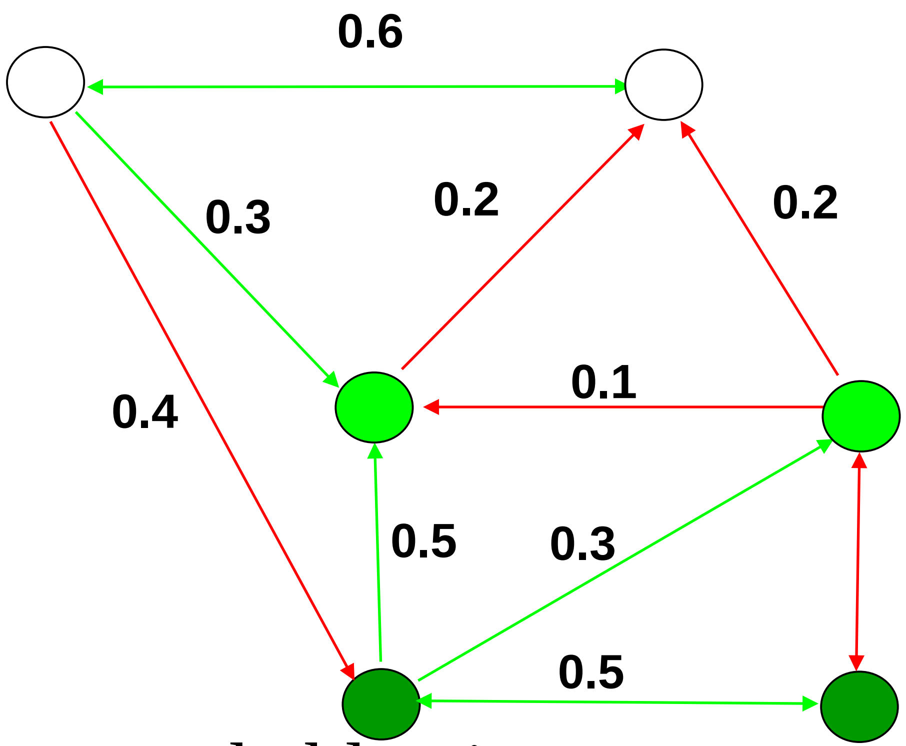
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.



Submodularity for Independent Cascade

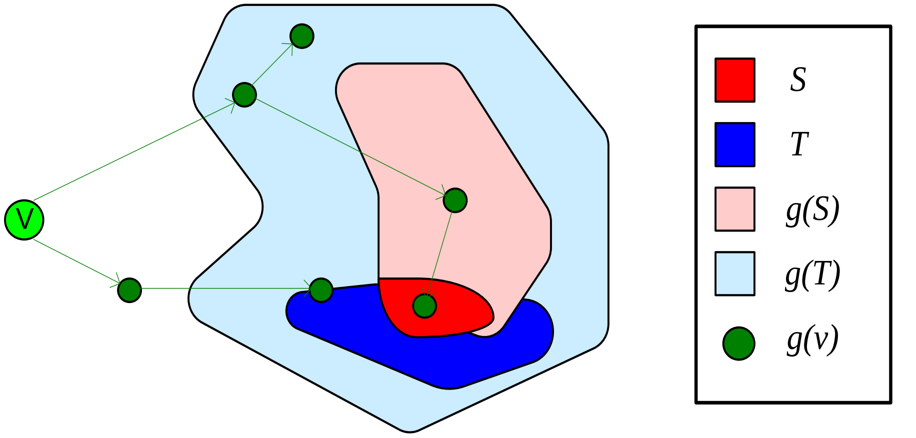
- Coins for edges are flipped during activation attempts.
- Can pre-flip all coins and reveal results immediately.



- Active nodes in the end are reachable via green paths from initially targeted nodes.
- Study reachability in green graphs

Submodularity, Fixed Graph

- Fix “green graph” G. $g(S)$ are nodes reachable from S in G.
- Submodularity: $g(T+v) - g(T) \subseteq g(S+v) - g(S)$ when $S \subseteq T$.



- $g(S+v) - g(S)$: nodes reachable from $S + v$, but not from S.
- From the picture: $g(T+v) - g(T) \subseteq g(S+v) - g(S)$ when $S \subseteq T$ *(indeed!)*.

Submodularity of the Function

Fact: A non-negative linear combination of submodular functions is submodular

$$f(S) = \sum_{G} \mathrm{Prob}(G \text{ } is \text{ } green \text{ } graph) \cdot g_G(S)$$

- $g_G(S)$: nodes reachable from S in G.
- Each $g_G(S)$: is submodular (previous slide).
- Probabilities are non-negative.

Submodularity for Linear Threshold

- Use similar "green graph" idea.
- Once a graph is fixed, "reachability" argument is identical.
- How do we fix a green graph now?
- Each node picks at most one incoming edge, with probabilities proportional to edge weights.
- Equivalent to linear threshold model (trickier proof).

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Key 2: Evaluating $f(S)$

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 shows: Greedy algorithm is now a $(1-1/e- \varepsilon')$-approximation.

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Experiment Data

- A collaboration graph obtained from co-authorships in papers of the arXiv high-energy physics theory section
- co-authorship networks arguably capture many of the key features of social networks more generally
- Resulting graph: 10748 nodes, 53000 distinct edges

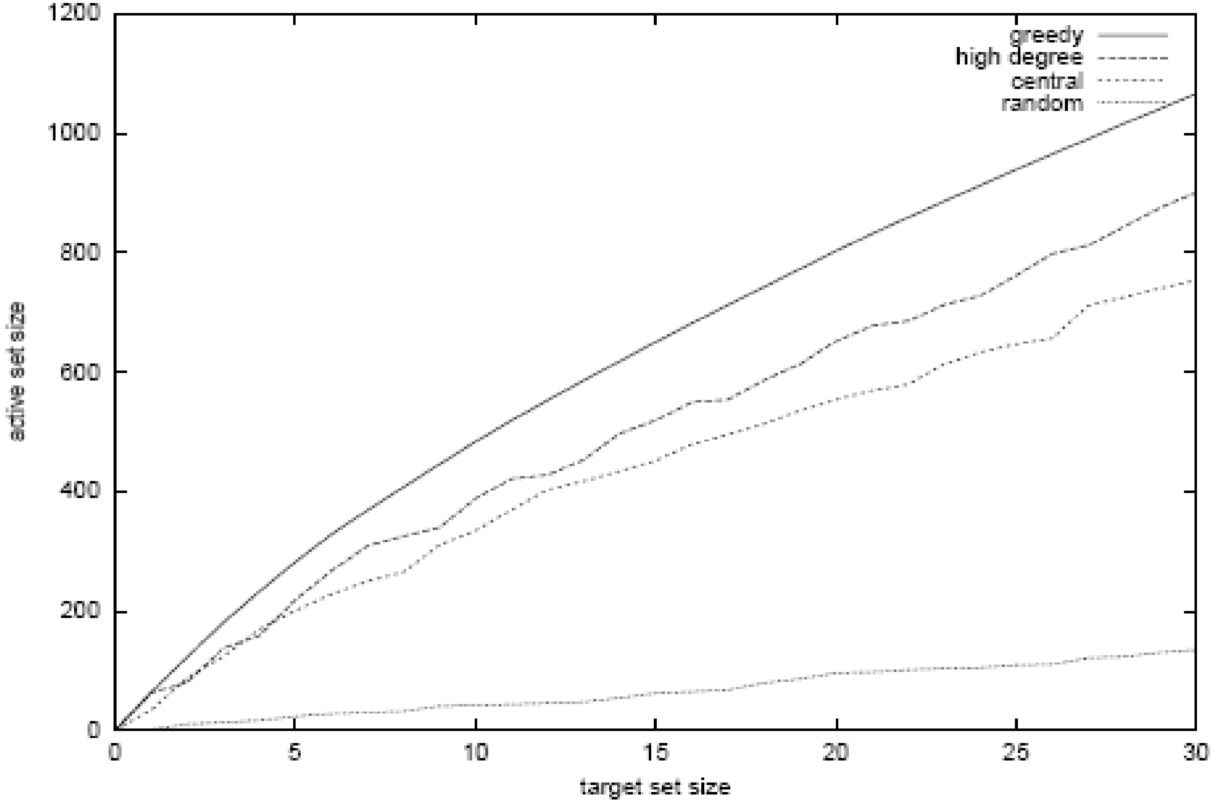
Experiment Settings

- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / *dv*, weight(ω→v) = C_{wv} / *dw*
- Independent Cascade Model:
  - Case 1: uniform probabilities *p* on each edge
  - Case 2: edge from *v* to *ω* has probability *1/ dω* of activating *ω*.
- Simulate the process 10000 times for each targeted set, re-choosing thresholds or edge outcomes pseudo-randomly from [0, 1] every time
- Compare with other 3 common heuristics
  - (in)degree centrality, distance centrality, random nodes.

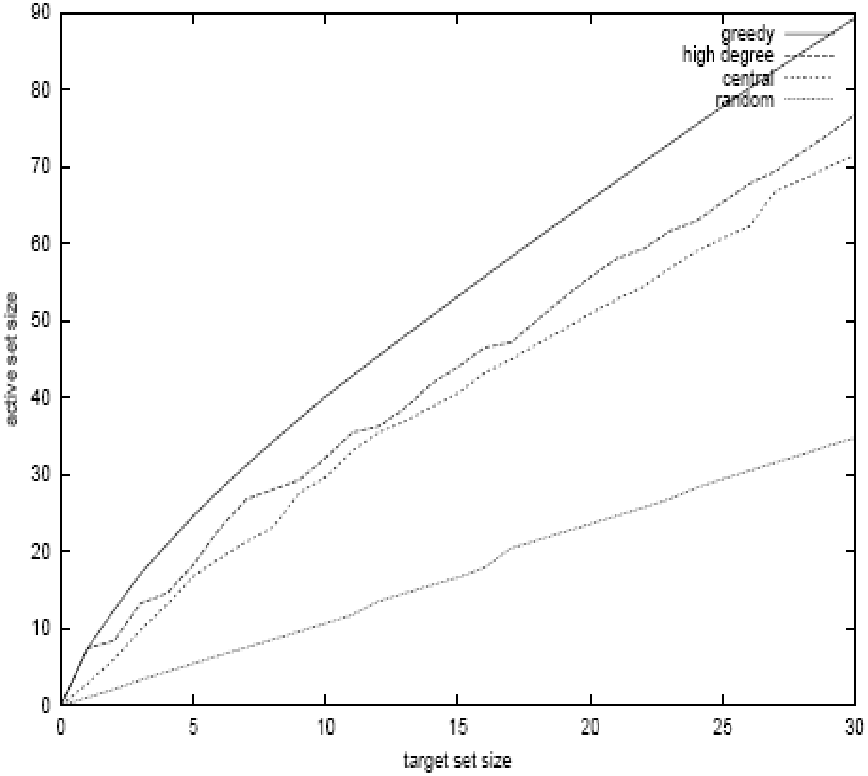
Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

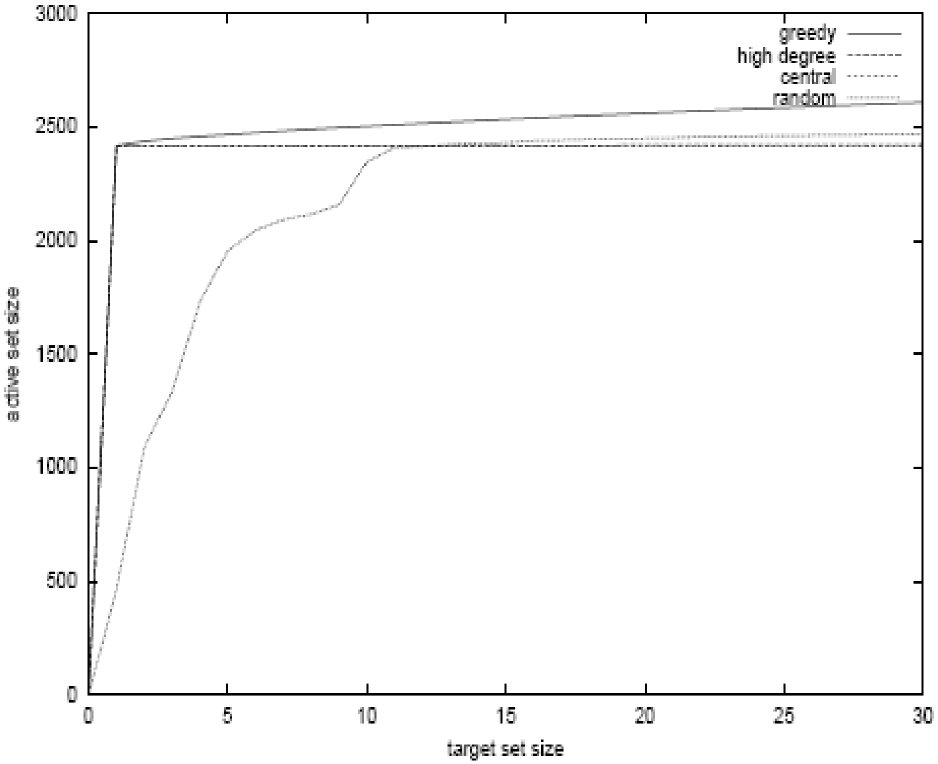
Results: linear threshold model



Independent Cascade Model – Case 1

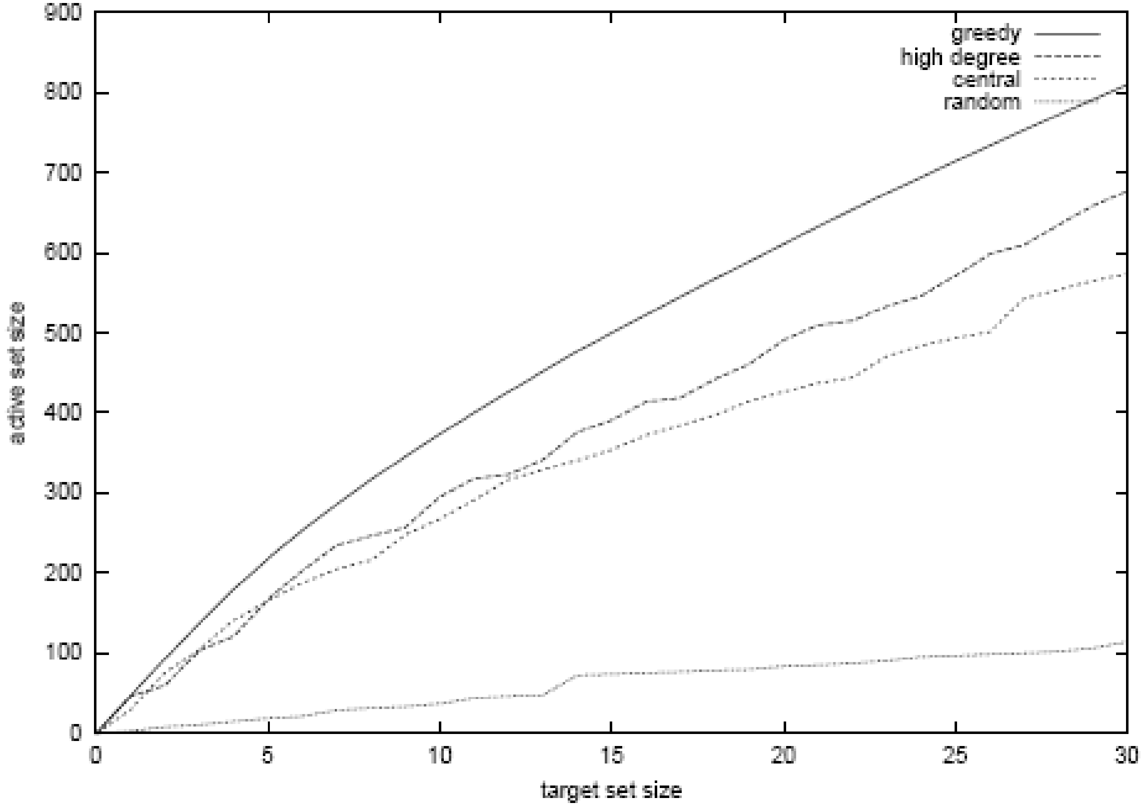


P = 1%



P = 10%

Independent Cascade Model – Case 2



Reminder: linear threshold model