Influence Maximization in Networks

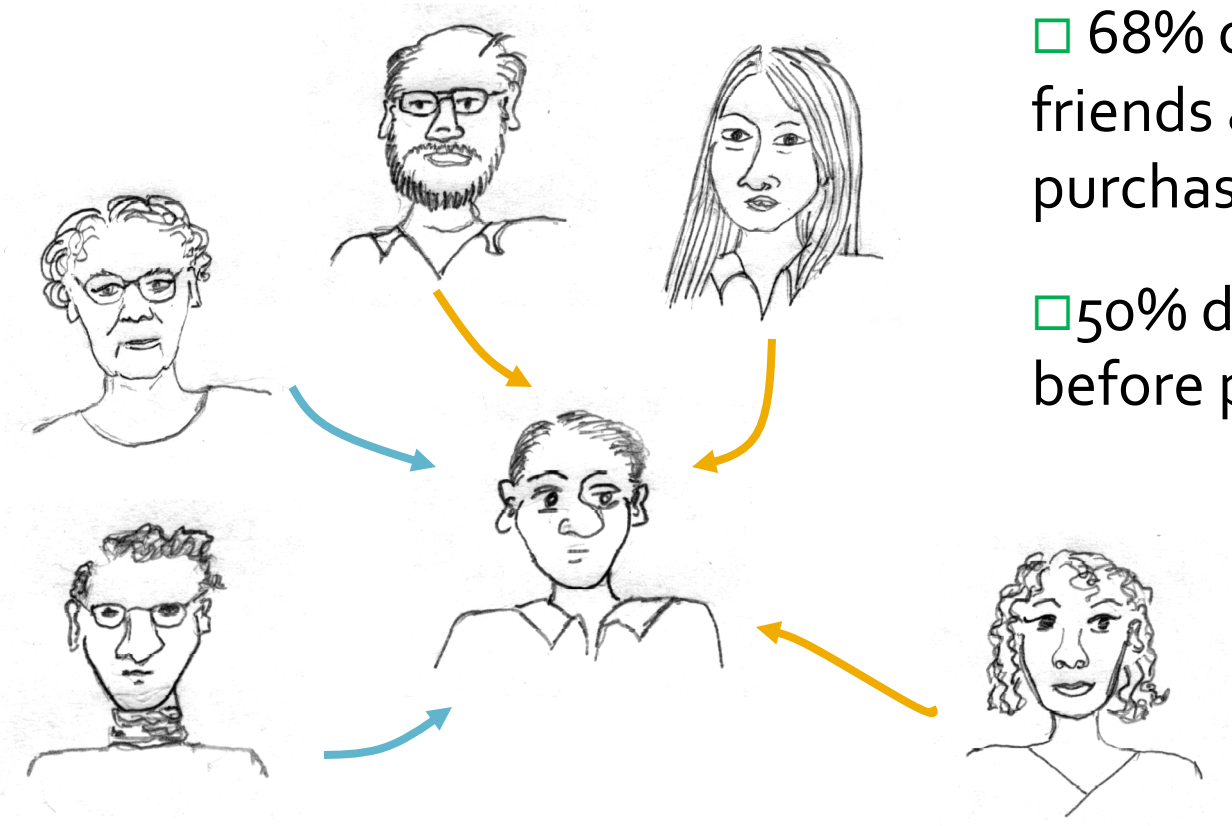
CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Viral Marketing?

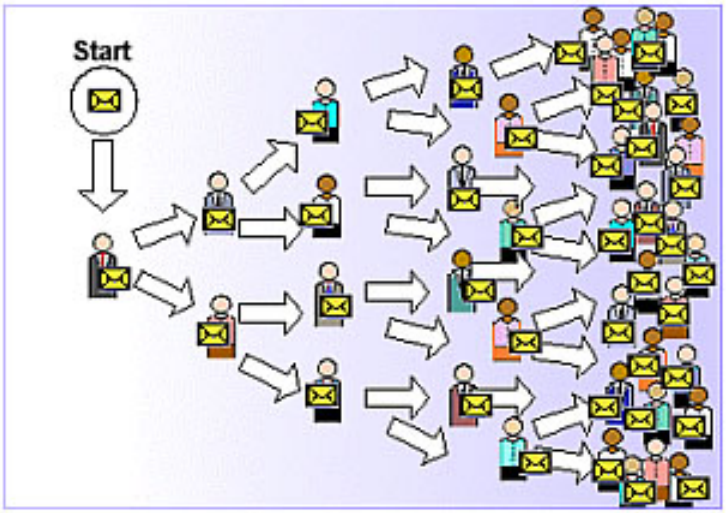
- **We are more influenced by our friends than strangers**



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

Viral Marketing

Identify influential customers



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

These customers endorse the product among their friends

Kate Middleton effect

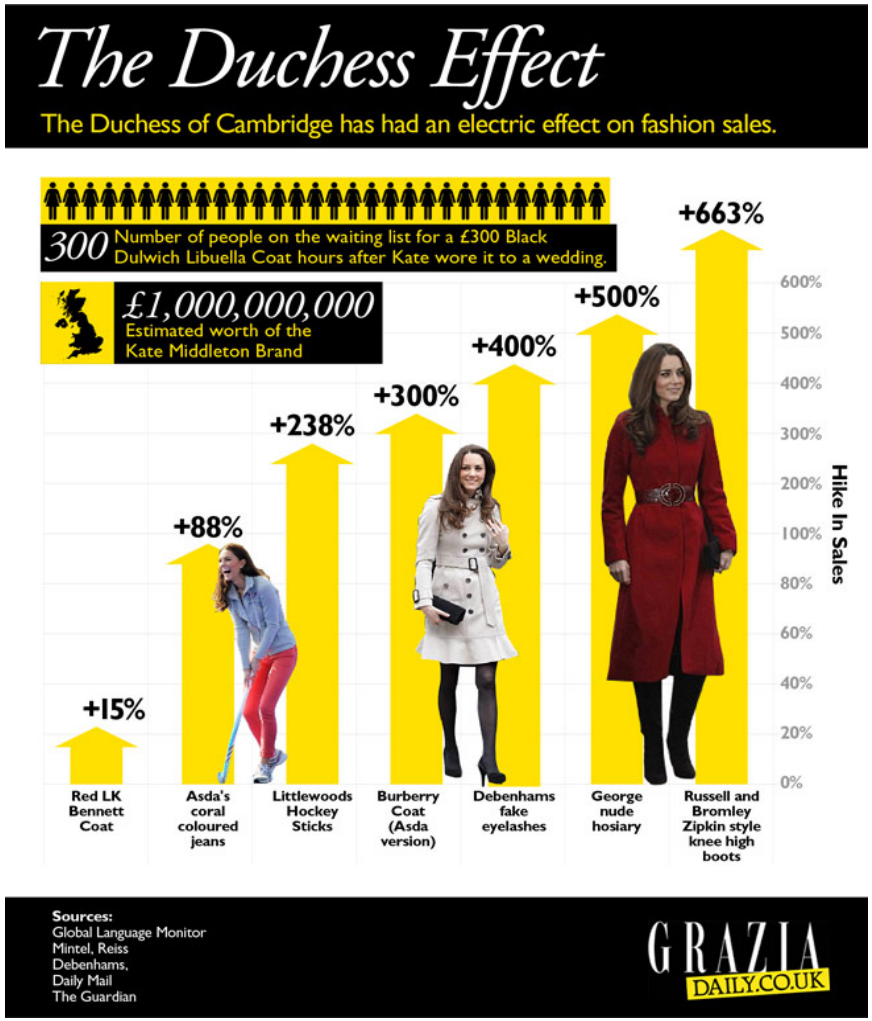
"Kate Middleton effect

The trend effect that Kate, Duchess of Cambridge has on others, from cosmetic surgery for brides, to sales of coral-colored jeans."

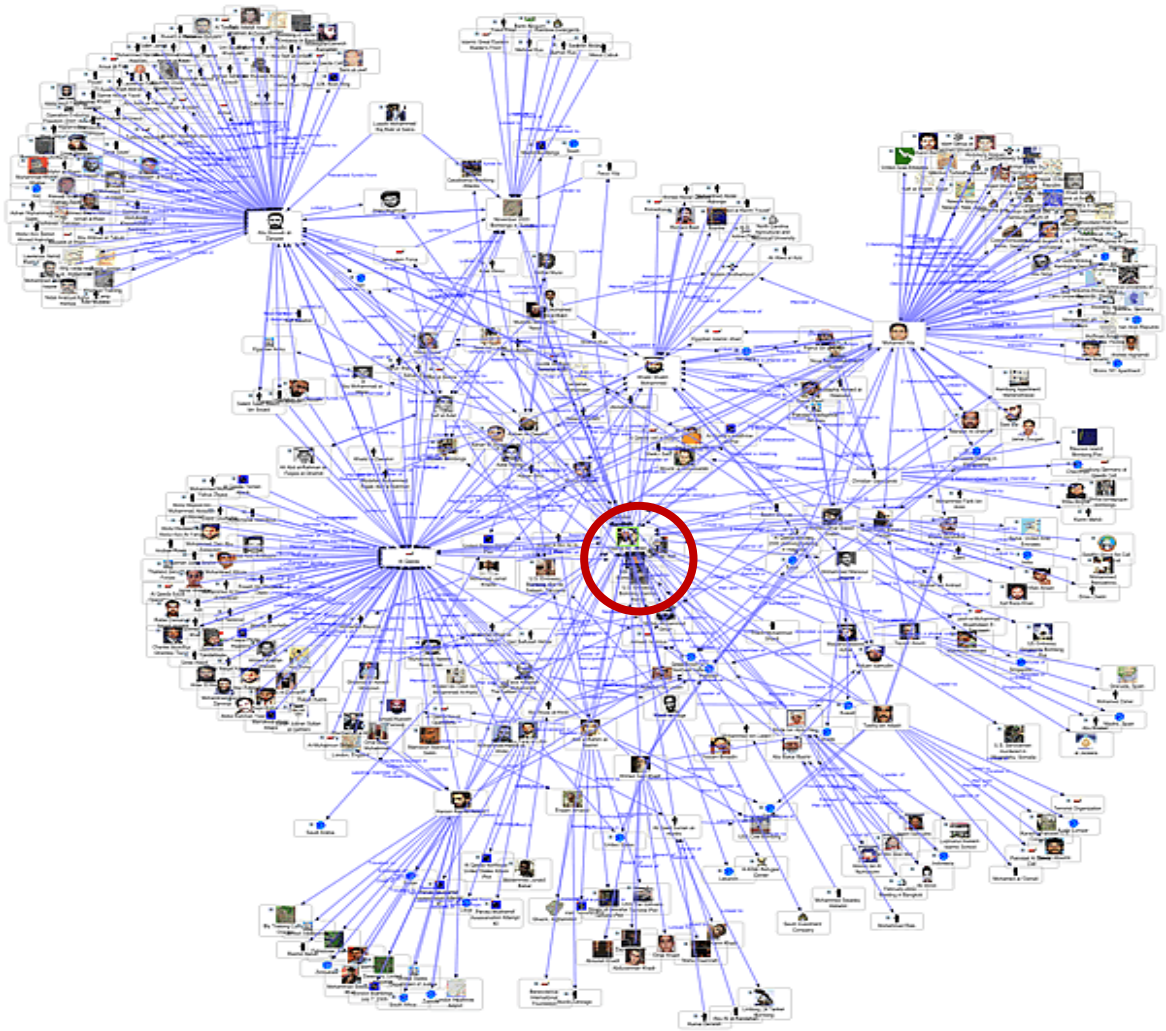
WIKIPEDIA
The Free Encyclopedia

Hike in Sales of Special Products

- According to Newsweek, "The Kate Effect may be worth £1 billion to the UK fashion industry."
- Tony DiMasso, L. K. Bennett's US president, stated in 2012, "...when she does wear something, it always seems to go on a waiting list."

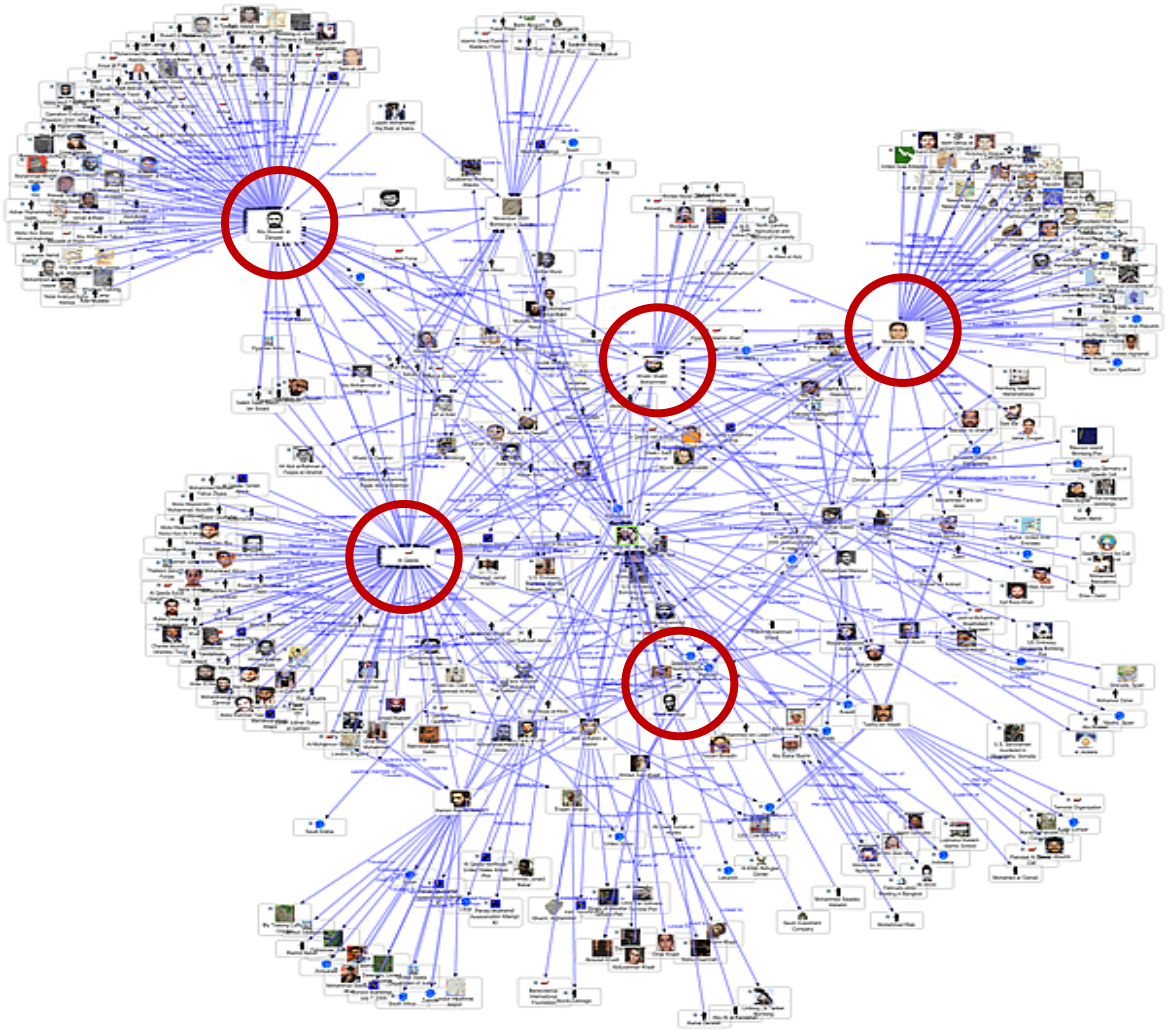


How to Find Kate?



- Influential persons often have many friends
- Kate is one of the persons that have many friends in this social network
- For more Kates, it's not as easy as you might think!

Influence Maximization



- Given a directed graph and *k>0,*
- Find *k* seeds (Kates) to maximize the number of influenced people (possibly in many steps)

Two Classical Propagation Models

- Linear Threshold Model
- Independent Cascade Model

Linear Threshold Model

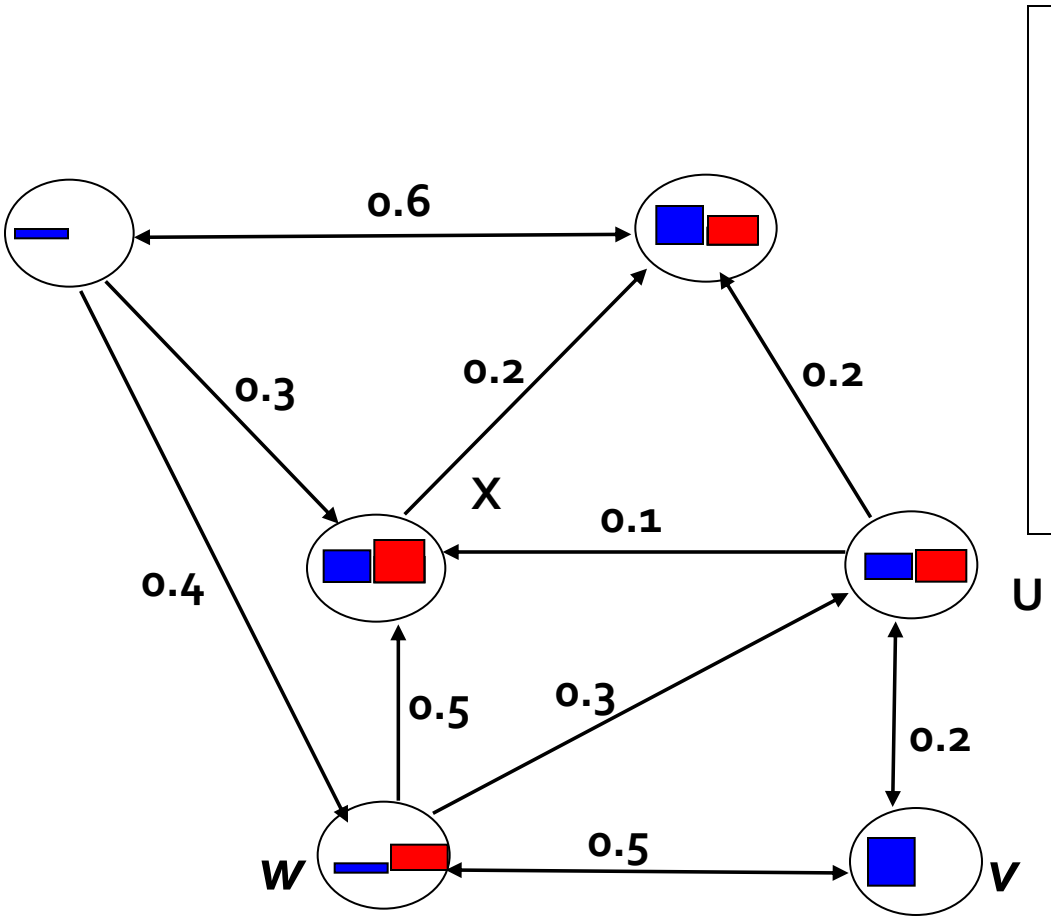
- A node *v* has random threshold $\theta_v \sim U[0,1]$
- A node *v* is influenced by each neighbor *w* according to a *weight* $b_{v,w}$ such that

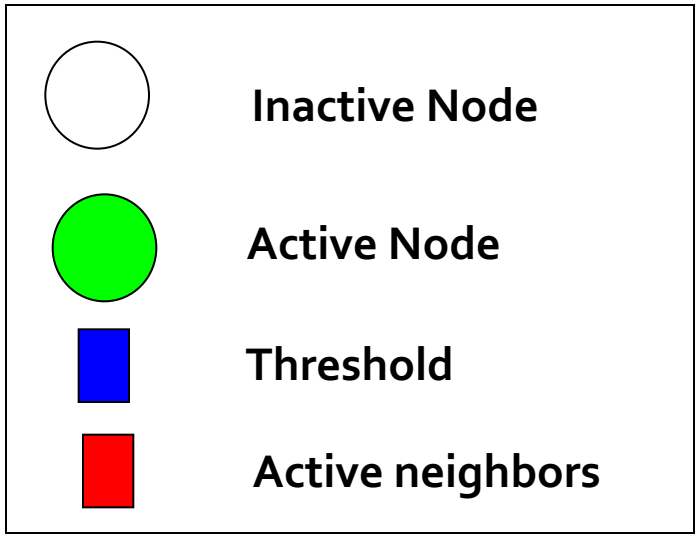
$$\sum_{w \text{ neighbor of } v} b_{v,w} \leq 1$$

- A node *v* becomes active when at least (weighted) θ_v fraction of its neighbors are active

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

Linear Threshold Model





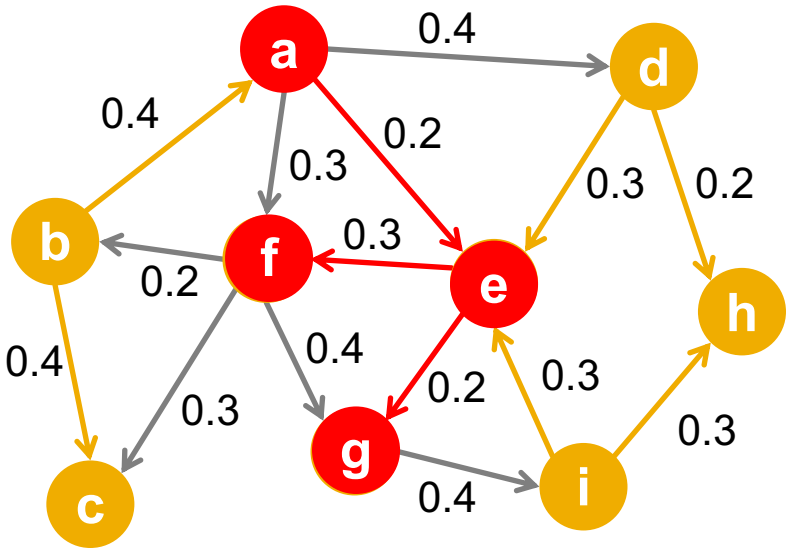
Stop!

Probabilistic Contagion

- **Independent Cascade Model**
  - Directed finite $\boldsymbol{G = (V, E)}$
  - Set $\boldsymbol{S}$ starts out with new behavior
    - Say nodes with this behavior are "**active**"
  - Each edge $\boldsymbol{(v, w)}$ has a probability $\boldsymbol{p_{vw}}$
  - If node $\boldsymbol{v}$ is active, it gets **one** chance to make $\boldsymbol{w}$ active, with probability $\boldsymbol{p_{vw}}$
    - Each edge fires at most once
- **Does scheduling matter? No**
  - If $\boldsymbol{u, v}$ are both active at the same time, it doesn't matter which tries to activate $\boldsymbol{w}$ 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}

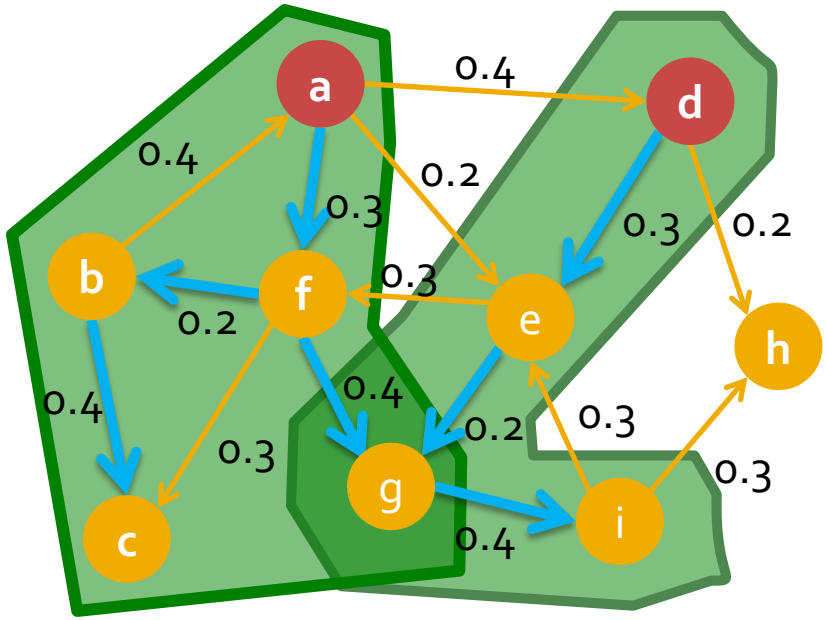


- **When node *v* becomes active:**
  - It activates each out-neighbor w with prob. p_{vw}
- **Activations spread through the network**

Most Influential Set

Problem: (*k* is a user-specified parameter)

- **Most influential set of size *k*:** set ***S*** of ***k*** nodes producing largest **expected** cascade size ***f(S)*** if activated [Domingos-Richardson '01]
- **Optimization problem:** $\max_{S \text{ of size } k} f(S)$



*Influence set X_a of **a***

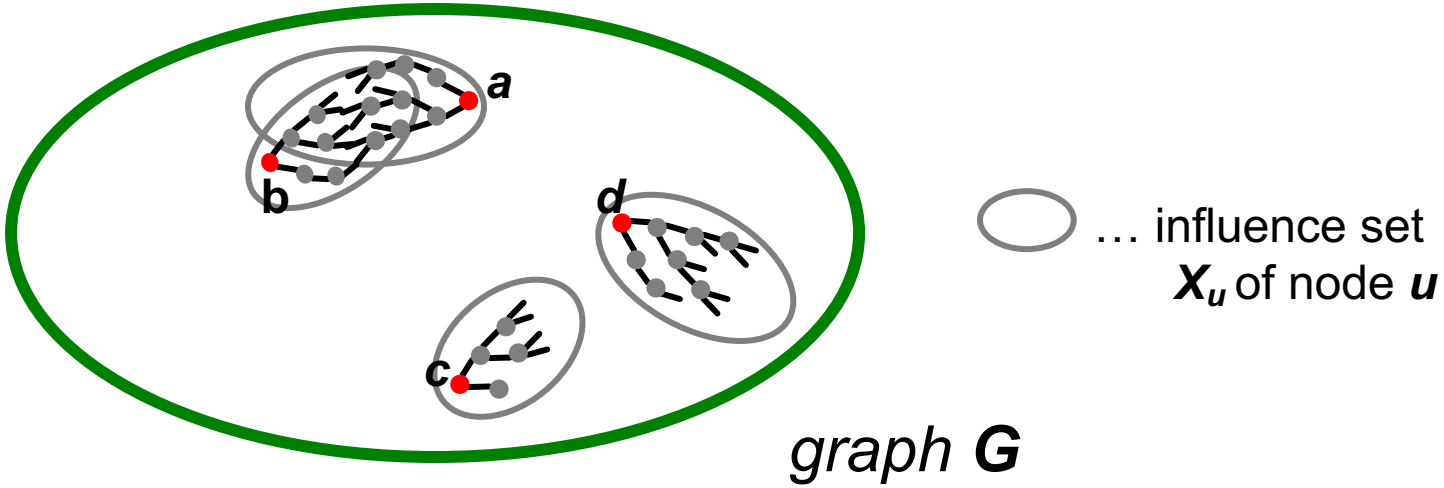
*Influence set X_d of **d***

Why "expected cascade size"? X_a is a result of a random process. So in practice we would want to compute X_a for many random realizations and then maximize the "average" value *f(S)*. For now let's ignore this nuisance and simply assume that each node *u* influences *a* set of nodes X_u

$$f(S) = \frac{1}{|I|} \sum_{\text{Random realizations } i} f_i(S)$$

Most Influential Set of Nodes

- ***S***: is initial active set
- ***f(S)***: **The expected size of final active set**
  - ***f(S)*** is the size of the union of X_u: $f(S) = |\cup_{u \in S} X_u|$



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

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

How hard is influence maximization?

Most Influential Subset of Nodes

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

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

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

Summary so Far

- **Extremely bad news:**
  - **Influence maximization is NP-complete**
- **Next, good news:**
  - **There exists an approximation algorithm!**
    - For some inputs the algorithm won't find globally optimal solution/set ***OPT***
    - But we will also prove that the algorithm will never do too badly either. More precisely, the algorithm will find a set $\boldsymbol{S}$ that where $\boldsymbol{f(S) \geq 0.63 * f(OPT)}$, where *OPT* is the globally optimal set.

The Approximation Algorithm

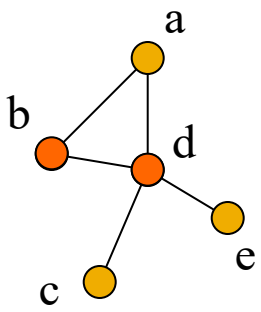
- **Consider a Greedy Hill Climbing algorithm to find S:**
  - **Input:**
    Influence set X_u of each node u: $X_u = \{v_1, v_2, \ldots\}$
    - That is, if we activate u, nodes $\{v_1, v_2, \ldots\}$ will eventually get active
  - **Algorithm:** At each iteration i activate the node u that gives largest 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, $u \in S_i$

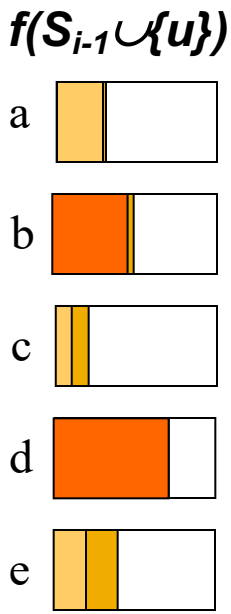
(Greedy) Hill Climbing

Algorithm:

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



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



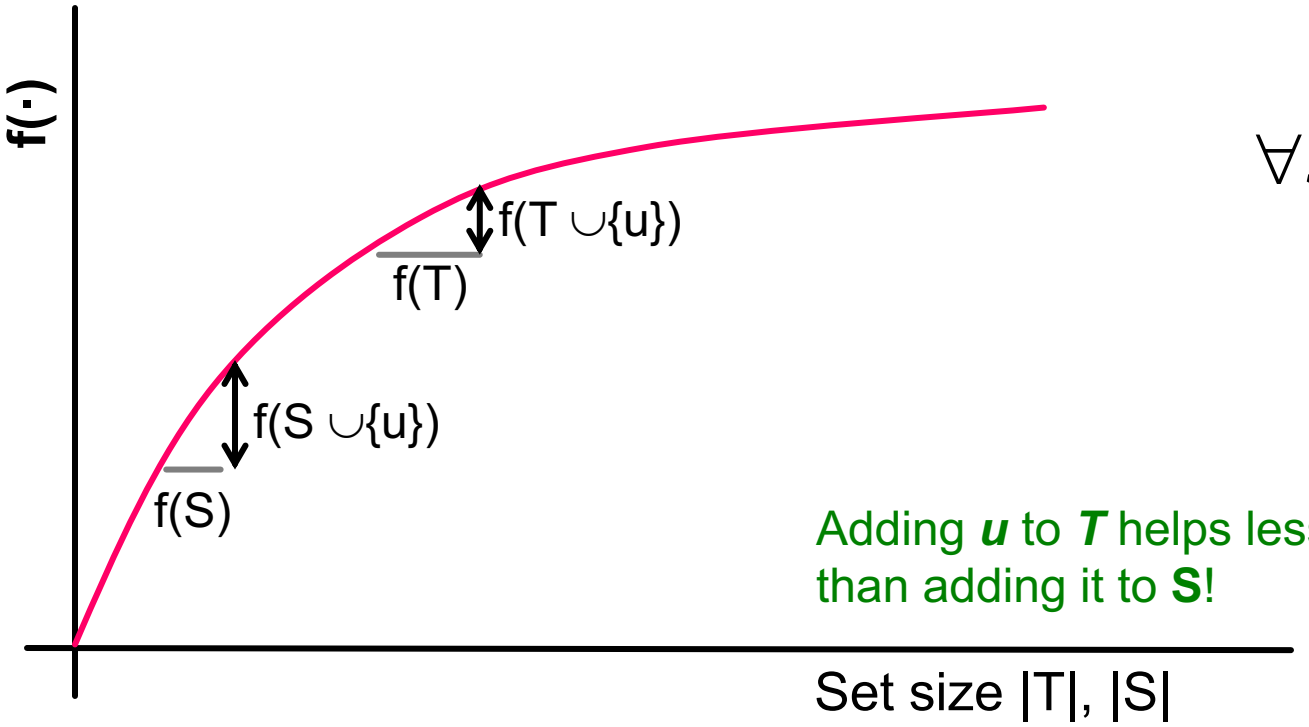
Approximation Guarantee

- **Claim: Hill climbing produces a solution S where: $f(S) \geq (1-1/e)*f(OPT)$** ($f(S)>0.63*f(OPT)$)
  [Nemhauser, Fisher, Wolsey '78, Kempe, Kleinberg, Tardos '03]
- **Claim holds for functions $f(\cdot)$ 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: $\forall S \subseteq T$

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

Submodularity– Diminishing returns

- **Diminishing returns:**



$$\forall S \subseteq T$$

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

Plan: Prove 2 things

(1) Our f(S) is submodular

(2) Hill Climbing gives near-optimal solutions

(for monotone submodular functions)

Also see the handout posted on the course website.

Background: Submodular Functions

- **We must show our $f(\cdot)$ is submodular:**
- $\forall S \subseteq T$

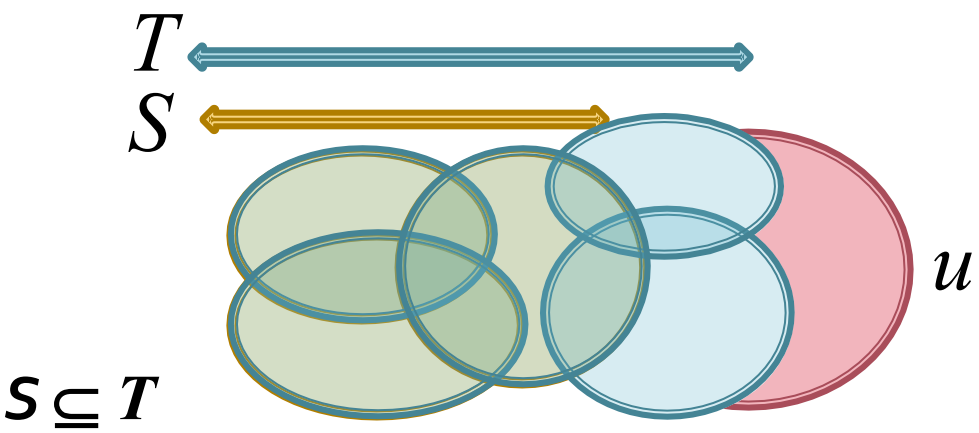
$$\underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding a node to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding a node to a large set}}$$

- **Basic fact 1:**
  - If $f_1(x), \ldots, f_k(x)$ are **submodular**, and $c_1, \ldots, c_k \geq 0$ then $F(x) = \sum_i c_i \cdot f_i(x)$ is also **submodular**

    (Non-negative combination of submodular functions is a submodular function)

Background: Submodular Functions

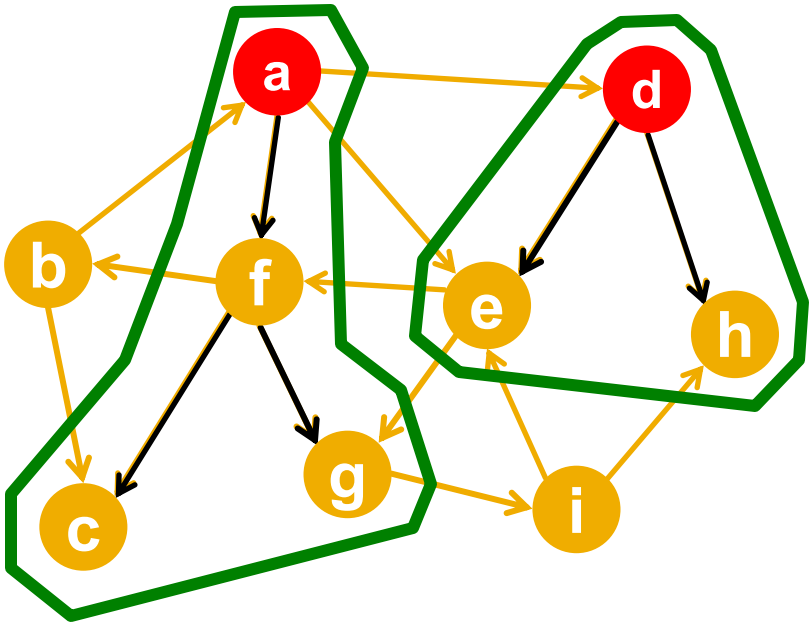
- $\forall S \subseteq T: \underbrace{f(S \cup \{u\}) - f(S)}_{\text{Gain of adding } u \text{ to a small set}} \geq \underbrace{f(T \cup \{u\}) - f(T)}_{\text{Gain of adding } u \text{ to a large set}}$
- **Basic fact 2: A simple submodular function**
  - Sets $X_1, \ldots, X_m$
  - $f(S) = |\cup_{k \in S} X_k|$ (size of the union of sets X_k, $k \in S$)
  - **Claim: $f(S)$ is submodular!**



The more sets you already have the less new area a given set ***u*** will cover

Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|}\sum_{\text{Random realizations } i} f_i(S)$$

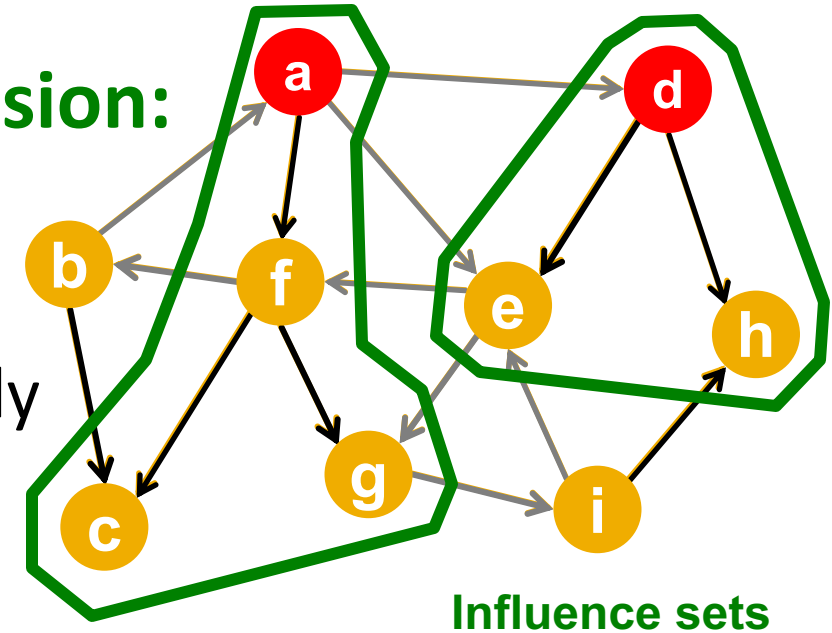


- **Proof strategy:**
  - We will argue that influence maximization is an instance of the **Set cover problem**:
    - ***Set cover problem:***
      f(S) is the size of the union of nodes influenced by active set **S**
      - Note ***f(S)*** is "random" (a result of a random process) so we need to be a bit careful
    - **Principle of deferred decision to the rescue!**
- We will create many **parallel possible worlds** and then average over them

Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|} \sum_{\text{Random realizations } i} f_i(S)$$

- **Principle of deferred decision:**
  - Flip all the coins at the beginning and record which edges fire successfully
  - **Now we have a deterministic graph!**
  - **Def:** Edge is **live** if it fired successfully
    - That is, we remove edges that did not fire



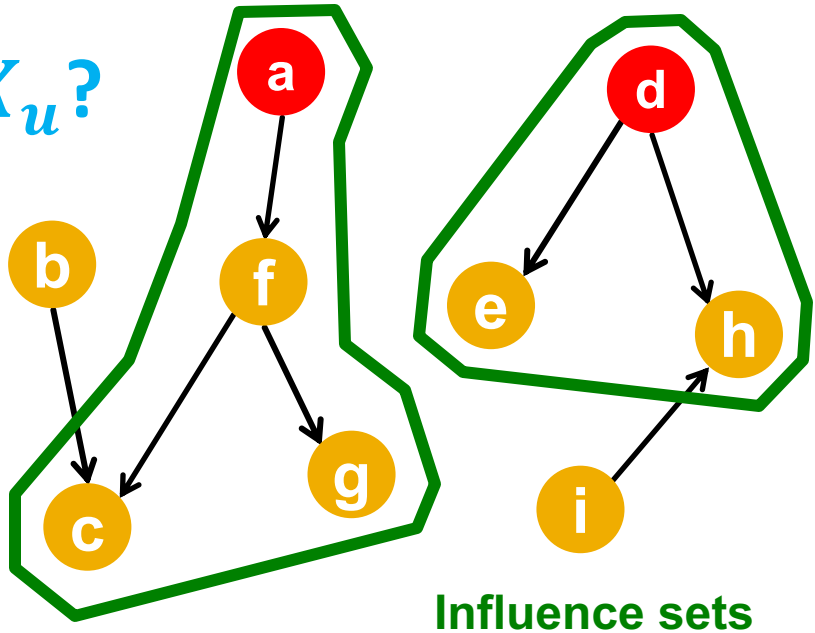
Influence sets for realization i:
$X_a^i = \{a,f,c,g\}$
$X_b^i = \{b,c\}$,
$X_c^i = \{c\}$
$X_d^i = \{d,e,h\}$
...

- **What is influence set X_u of node u?**
  - **The set reachable by live-edge paths from u**

Our $f(S)$ is Submodular!

$$f(S) = \frac{1}{|I|} \sum_{\text{Random realizations } i} f_i(S)$$

- What is an influence set X_u?
  - The set reachable by live-edge paths from u
- What is now $f(S)$?
  - $f_i(S)$ = size of the set reachable by live-edge paths from nodes in S



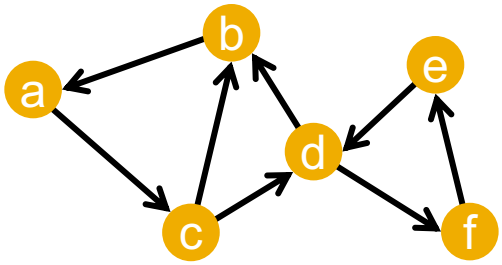
Influence sets for realization i:

X_a^i = {a,f,c,g}
X_b^i = {b,c},
X_c^i = {c}
X_d^i = {d,e,h}

- For the i-th possible world (realization of coin flips)
  - $f_i(S = \{a, b\}) = |\{a, f, c, g\} \cup \{b, c\}| = 5$
  - $f_i(S = \{a, d\}) = |\{a, f, c, g\} \cup \{d, e, h\}| = 7$

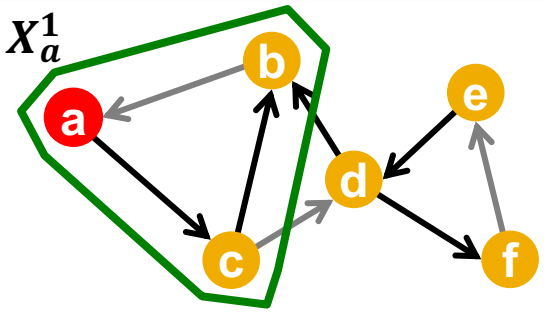
Our $f(S)$ is Submodular!

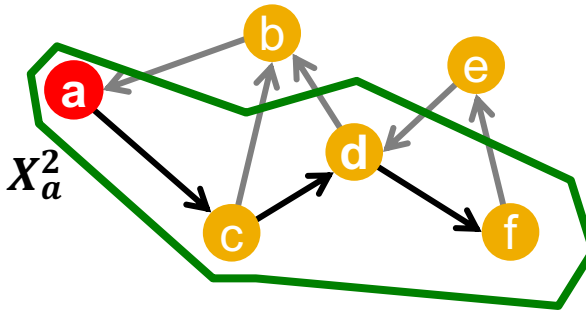
$$f(S) = \frac{1}{|I|}\sum_{\text{Random realizations } i} f_i(S)$$

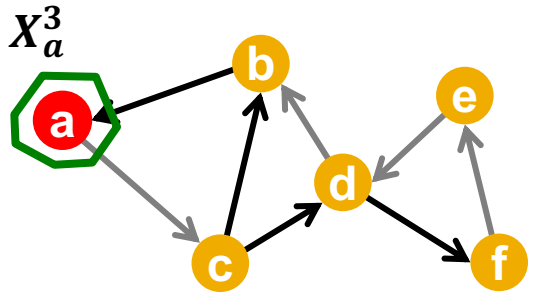


Activate edges by coin flipping

Possible worlds







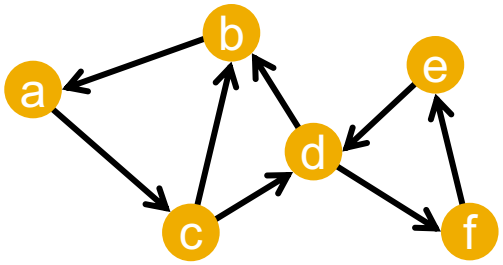
- **Generate a possible world:**
  - Fix outcome $i \in I$ of coin flips
- X_v^i = set of nodes reachable from v on live-edge paths
- $f_i(S)$ = size of cascades from S given the coin flips i
- $f_i(S) = \left|\bigcup_{v \in S} X_v^i\right| \Rightarrow f_i(S)$ is **submodular!**
  - X_v^i are sets, $f_i(S)$ is the size of their union
- **Expected influence set size:**

$$f(S) = \frac{1}{|I|}\sum_{i \in I} f_i(S) \Rightarrow f(S) \text{ is } \textbf{submodular!}$$

  - $f(S)$ is a linear combination of submodular functions

RECAP: Influence Maximization

- **Find most influential set *S* of size *k*:** largest expected cascade size ***f(S)*** if set ***S*** is activated

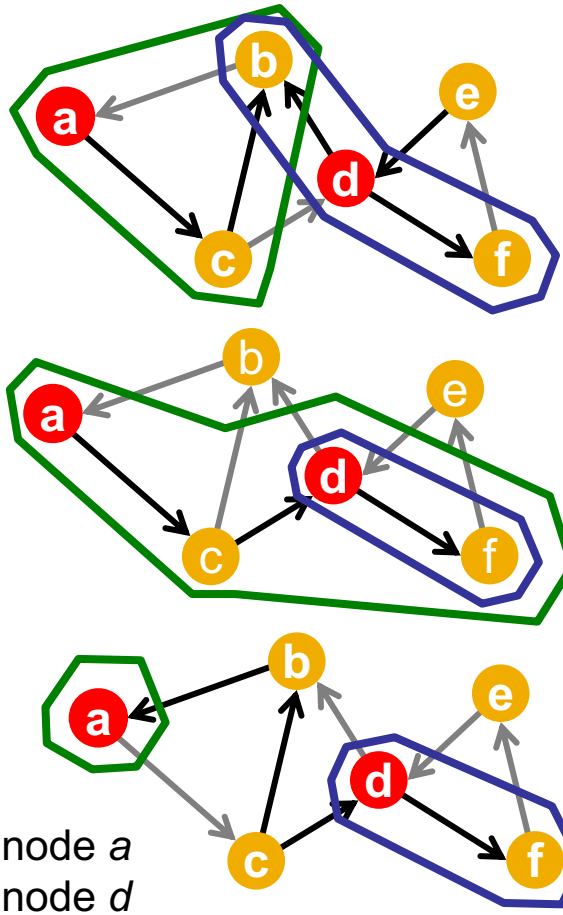


Network, each edge activates with prob. p_{uv}

Activate edges by coin flipping

Possible worlds

Multiple realizations ***i.***
Each realization is a "possible world"



- **Want to solve:**

$$\arg\max_{|S|=k} f(S) = \frac{1}{|I|}\sum_{i\in I} f_i(S)$$

Consider ***S={a,d}*** then:
$f_1(S)=5$, $f_2(S)=4$, $f_3(S)=3$
and **f(S) = 1/3*12=4**

… influence set of node *a*
… influence set of node *d*

Plan: Prove 2 things

(1) Our f(S) is submodular

(2) Hill Climbing gives near-optimal solutions

(for monotone submodular functions)

Proof for Hill Climbing

Claim:

When f(S) is monotone and submodular then Hill climbing produces active set S

where: $\boldsymbol{f(S) \geq \left(1 - \frac{1}{e}\right) \cdot f(OPT)}$

- In other words: $f(S) \geq 0.63 \cdot f(OPT)$

- **The setting:**
  - Keep adding nodes that give the largest gain
  - Start with $\boldsymbol{S_0 = \{\}}$, produce sets $\boldsymbol{S_1, S_2, \ldots, S_k}$
  - **Add elements one by one**
  - Let $\boldsymbol{OPT = \{t_1 \ldots t_k\}}$ be **the optimal set (OPT)** of size $\boldsymbol{k}$
- **We need to show:** $\boldsymbol{f(S) \geq (1 - \frac{1}{e})\, f(OPT)}$

(Do at home. See handout!)

Solution Quality

We just proved: ☺

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

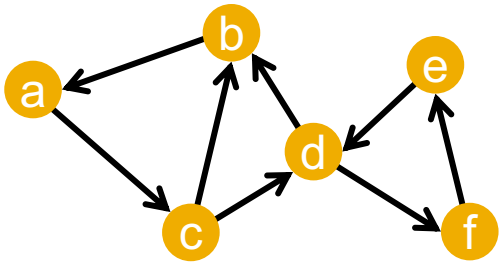
- This is a **data independent bound**
  - This is a worst case bound
  - No matter what is the input data, we know that the Hill-Climbing **will never do worse** than $0.63*f(OPT)$

Evaluating our $f(S)$?

- **How to evaluate influence maximization $f(S)$?**
  - Still an open question of how to compute it 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**

RECAP: Influence Maximization

- **Find most influential set *S* of size *k*:** largest expected cascade size ***f(S)*** if set ***S*** is activated

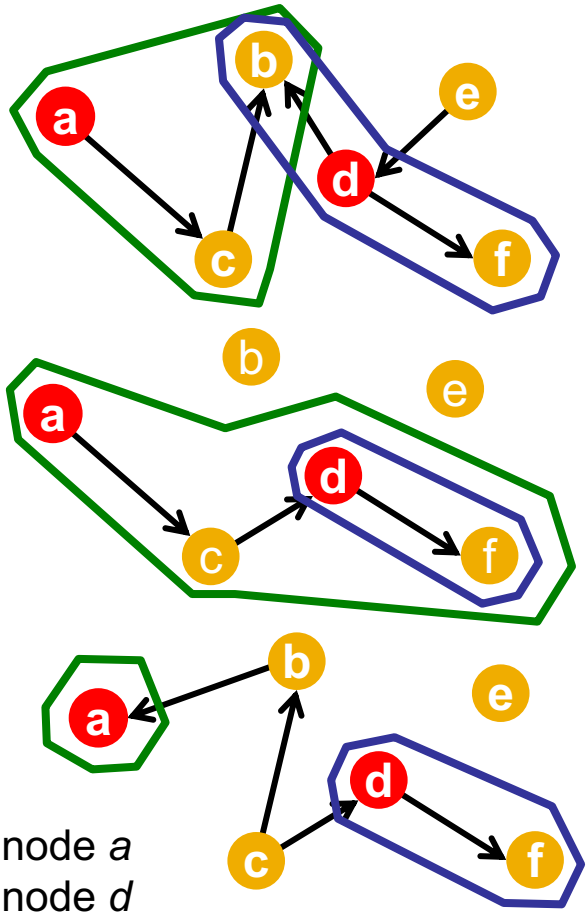


Network, each edge activates with prob. p_{uv}

Activate edges by coin flipping

Possible worlds

Multiple realizations ***i.***
Each realization is a "parallel universe"



- **Want to solve:**

$$\arg\max_{|S|=k} f(S) = \frac{1}{|I|}\sum_{i\in I} f_i(S)$$

Consider ***S={a,d}*** then:
$f_1(S)=5$, $f_2(S)=4$, $f_3(S)=3$
and **f(S) = 1/3*(5+4+3)=4**

… influence set of node *a*
… influence set of node *d*

Greedy Algorithm is Slow

- **Notice:** Greedy approach is slow!
  - For a given network ***G***, repeat 10,000s of times:
    - Flip coin for each edge and determine influence sets under coin-flip realization *i*
    - Each node *u* is associated with 10,000s influence sets X_u^i
  - **Greedy's complexity is $O(k \cdot n \cdot R \cdot m)$**
    - n … number of nodes in *G*
    - k … number of nodes to be selected/influenced
    - R … number of simulation rounds (number possible worlds)
    - m … number of edges in *G*

Experiments and Concluding Thoughts

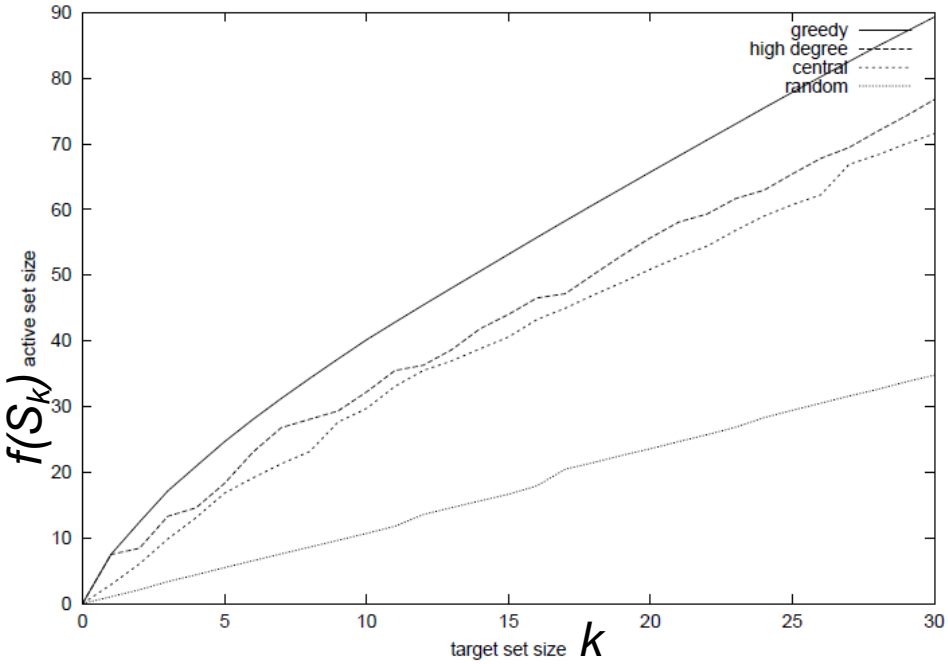
Experiment Data

- **A collaboration network**: co-authorships in papers of the arXiv high-energy physics theory:
  - 10,748 nodes, 53,000 edges
  - **Example cascade process:** Spread of new scientific terminology/method or new research area
- **Independent Cascade Model:**
  - **Each user's threshold is uniform random on [0,1]**
  - **Case 1:** Uniform probability ***p*** on each edge
  - **Case 2:** Edge from **v** to **w** has probability **1/deg(w)** of activating **w**.

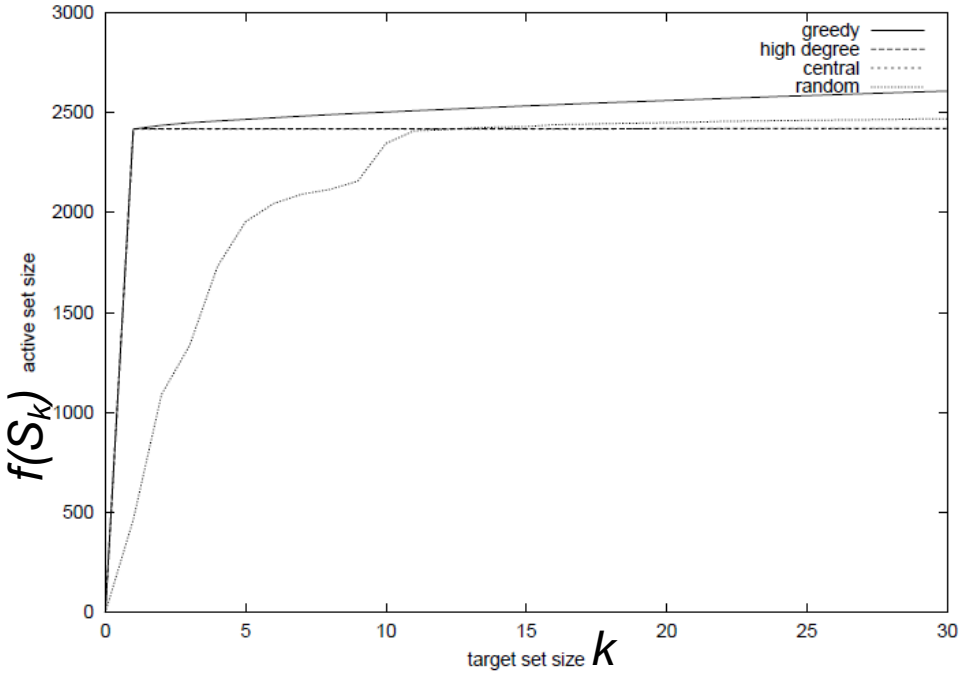
Experiment Settings

- **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:** Pick nodes with highest degree
  - **Closeness centrality:** Pick nodes in the "center" of the network
  - **Random nodes:** Pick a random set of nodes

Independent Cascade Model



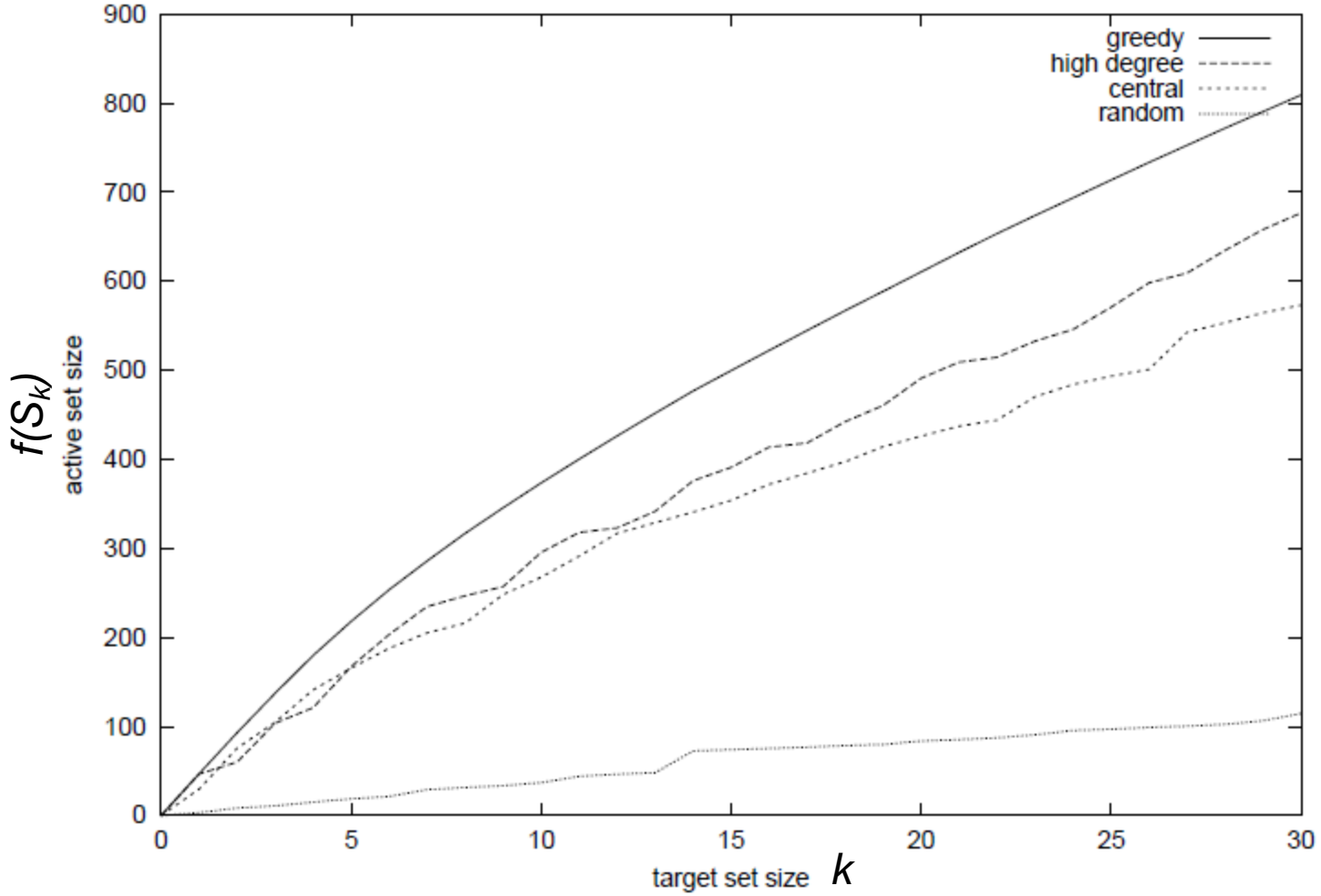
$p_{uv} = 0.01$



$p_{uv} = 0.10$

Uniform edge firing probability p_{uv}

Independent Cascade Model



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

Non-uniform edge firing probability p_{uv}

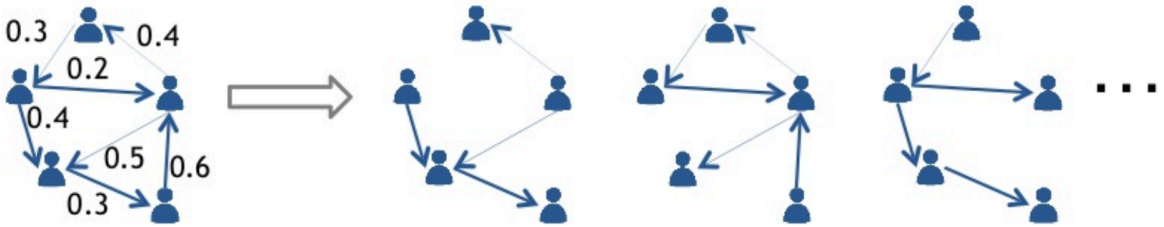
Speeding things up: Sketch-based Algorithms

Sketch-based Influence Maximization and Computation: Scaling up with Guarantees", CIKM 2014

Sketch-based Algorithms

To perform influence maximization we need to…

- 1) Generate a number R of possible worlds



- 2) Identify k nodes with the largest influence in these possible worlds
- **Problem:**
  - For any given node set, evaluating its influence in a possible world takes $O(m)$ time (m… number of edges)
- **Solution:** Use sketches to reduce estimation time from $O(m)$ to $O(1)$

Reachability Sketches

- **Solution:** Use sketches to reduce estimation time from $O(m)$ to $O(1)$

Idea:

- Compute small structure per node from which to estimate its influence

- Then run influence maximization using these estimates

Reachability Sketches

Rough Idea:

- Take a possible world $G^{(i)}$
- Give each node a uniform random number from [0,1]

- Compute the rank of each node v, which is the minimum number among the nodes that v can reach

Reachability Sketches

Intuition

- If v can reach a large number of nodes then its rank is likely to be small

- Hence, the rank of node v can be used to estimate the influence of node v a graph in a possible word $G^{(i)}$

Reachability Sketches

Problem

- Influence estimation based on a single rank/number can be inaccurate

Solution

- Keep multiple ranks/numbers
  - E.g., keep the smallest c values among the nodes that v can reach

Reachability Sketches

Problem

- Influence estimation based on single rank/number can be inaccurate

Solution

- Keep multiple ranks/numbers
  - E.g., keep the smallest c values among the nodes that v can reach
- Enables estimate on union of these reachable sets

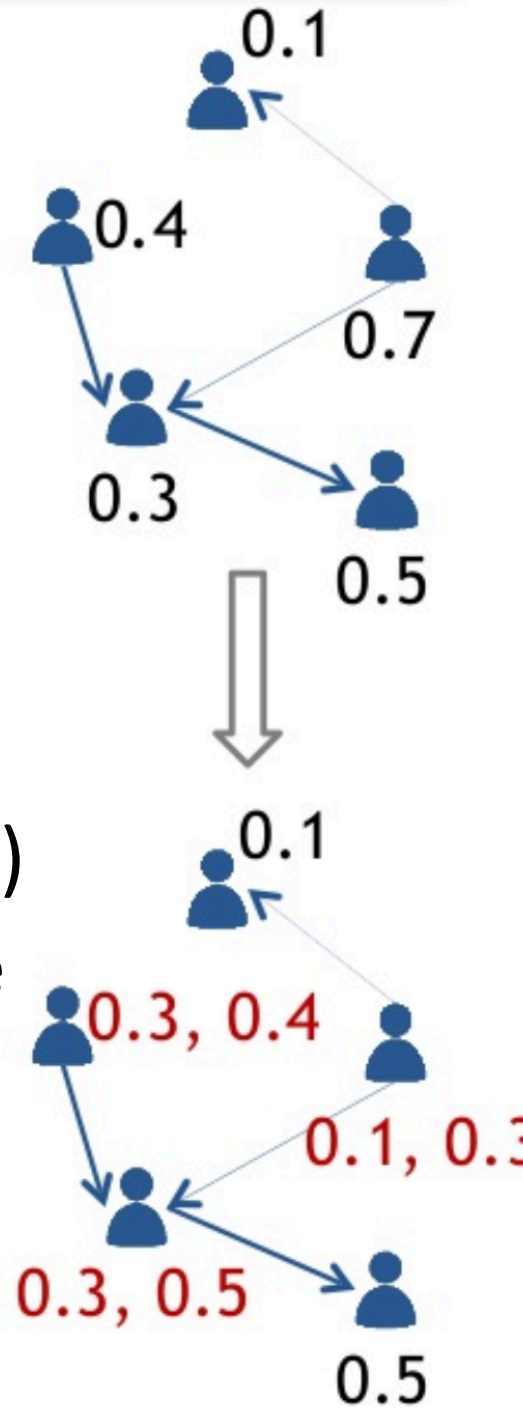
Reachability Sketches

Problem

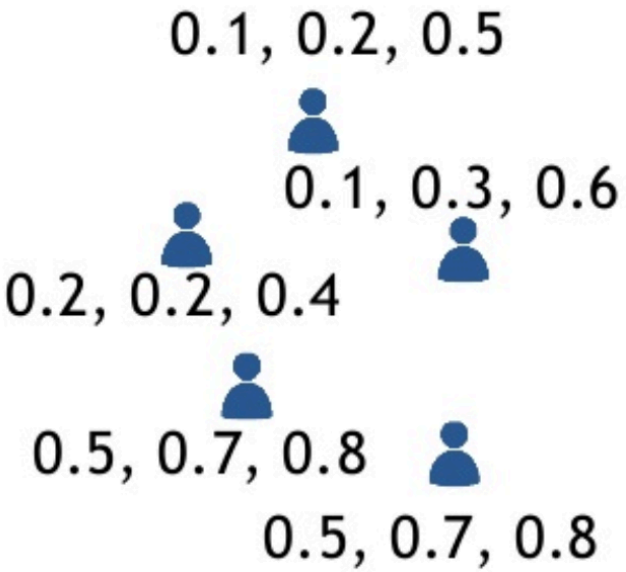
- Influence estimation based on single rank/number can be inaccurate

Solution

- Keep multiple ranks (say c of them)
  - Keep the smallest c values among the nodes that v can reach in all possible worlds considered (but keep the numbers fix across the worlds)



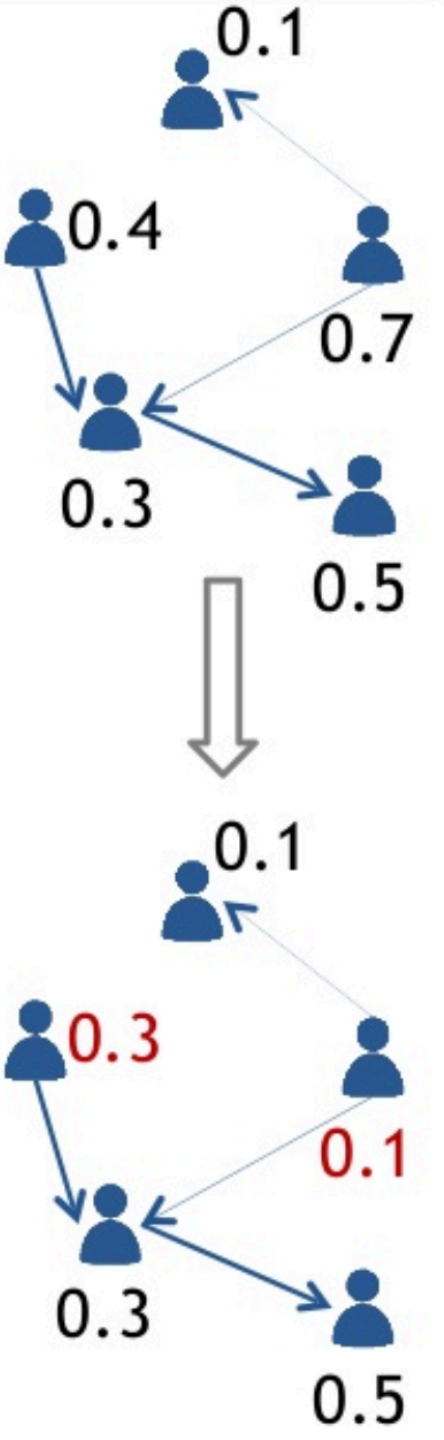
Sketch-based Greedy



- Generate a number of possible worlds
- Construct reachability sketches for all node:
  - **Result:** each node has c ranks
- **Run Greedy for influence maximization**
  - Whenever Greedy asks for the influence of a node set S, check ranks and add a u node that has the smallest value (lexicographically)
    - After u is chosen. Find its influence set of nodes $f(u)$, mark them as infected and remove their "numbers" from the sketches of other nodes

Sketch-based Greedy

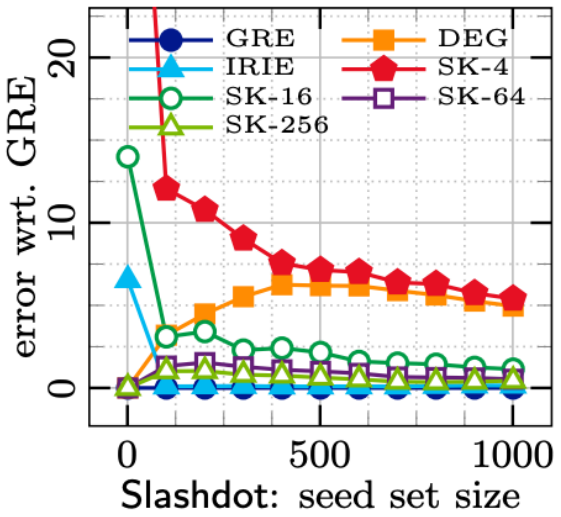
- **Guarantees:**
  - Expected running time is near-linear in the number of possible worlds
  - When c is large, it provides $(1 - \frac{1}{e} - \varepsilon)$ approx with respect to the possible worlds considered

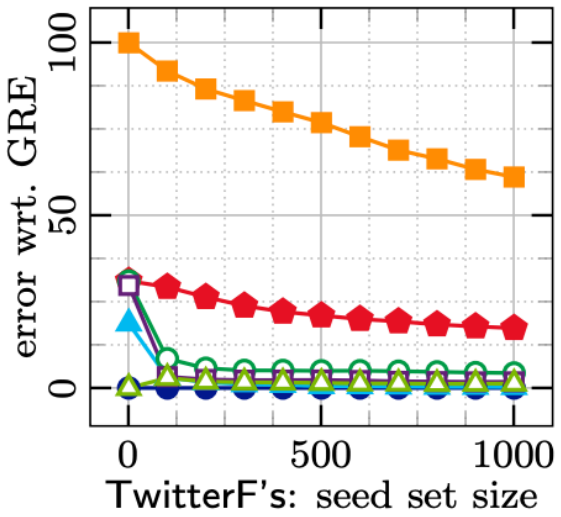


Summary

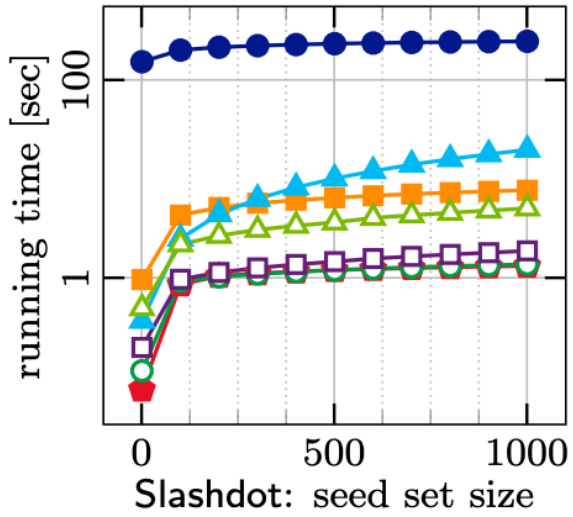
- **Advantages:**
  - Expected near-linear running time
  - Provides an approximation guarantee with respect to the possible worlds considered
- **Disadvantage**
  - Does not provide an approximation guarantee on the ”true” expected influence

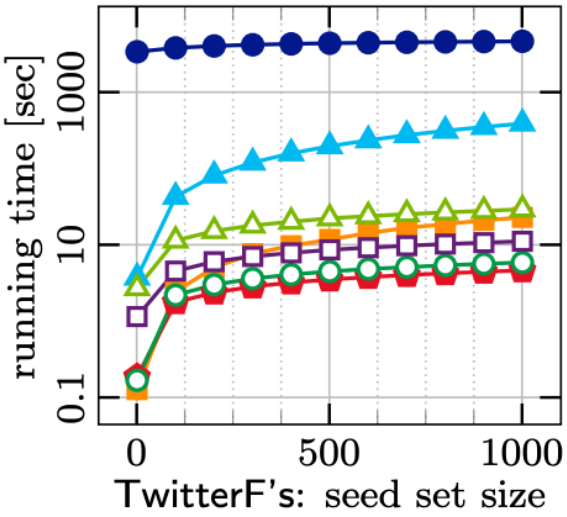
Experiments





GRE… greedy
IRIE… state of the art heuristics
DEG…degree based heuristics
SK... Sketch-based





Sketch-based achieves the same performance as greedy in a fraction of the time!

Open Questions

- **More realistic viral marketing:**
  - Different marketing actions increase **likelihood** of initial activation, for **several** nodes at once
- **Study more general influence models:**
  - Find trade-offs between generality and feasibility
- **Deal with negative influences:**
  - Model competing ideas
- Obtain more data (better models) about how activations occur in real social networks